SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 1–14

# SMOOTHING-NORM PRECONDITIONING FOR REGULARIZING MINIMUM-RESIDUAL METHODS*

PER CHRISTIAN HANSEN† AND TOKE KOLDBORG JENSEN†‡

**Abstract.** When GMRES (or a similar minimum-residual algorithm such as RRGMRES, MINRES, or MR-II) is applied to a discrete ill-posed problem with a square matrix, in some cases the iterates can be considered as regularized solutions. We show how to precondition these methods in such a way that the iterations take into account a smoothing norm for the solution. This technique is well established for CGLS, but it does not immediately carry over to minimum-residual methods when the smoothing norm is a seminorm or a Sobolev norm. We develop a new technique which works for any smoothing norm of the form $\|L\,x\|_2$ and which preserves symmetry if the coefficient matrix is symmetric. We also discuss the efficient implementation of our preconditioning technique, and we demonstrate its performance with numerical examples in one and two dimensions.

**Key words.** general-form regularization, smoothing norm, regularizing iterations, GMRES, MINRES, weighted pseudoinverse

**AMS subject classifications.** 65F22, 65F10

**DOI.** 10.1137/050628453

## 1. Introduction.

We are concerned with large-scale discrete ill-posed problems with a square coefficient matrix, i.e., ill-conditioned linear systems of the form $A\,x = b$ with $A \in \mathbb{R}^{n\times n}$ and $x,\, b \in \mathbb{R}^n$. These problems typically arise from discretizations of Fredholm integral equations of the first kind, e.g., in computerized tomography, geophysics, or image restoration. Due to the ill-conditioning of $A$ and the unavoidable errors in the right-hand side (coming from data), any attempt to compute the "naive" solution $A^{-1}b$ will fail to produce a meaningful solution.

Instead we must use a regularization method to compute a stabilized solution which is less sensitive to the errors. There are many such methods around, and one of the most popular is Tikhonov regularization, which amounts to computing

$$x_\lambda = \operatorname{argmin}_x \left\{ \|A\,x - b\|_2^2 + \lambda^2\,\|L\,x\|_2^2 \right\} = (A^TA + \lambda^2\,L^TL)^{-1}A^Tb, \tag{1.1}$$

where the matrix $L$ defines a [illegible] $\|L\cdot\|_2$ that acts as a regularizer and where $\lambda$ is the regularization parameter.

For large-scale problems we need iterative methods to compute regularized solutions, and there is a rich literature on CG-based methods for computing the Tikhonov solution via the least-squares formulation of (1.1). More recently we have seen an interest in methods referred to as [illegible] These are Krylov subspace methods applied directly to the problem $\min\|A\,x - b\|_2$ or $A\,x = b$ with no additional smoothing norm (such as $\lambda^2\|L\,x\|_2^2$); instead the projection of the problem onto the Krylov subspace, associated with the method, acts as a regularizer of the solution. See, e.g., [7] and [12] for details.

---

[*] Received by the editors April 4, 2005; accepted for publication (in revised form) by D. Szyld March 23, 2006; published electronically December 21, 2006.
http://www.siam.org/journals/simax/29-1/62845.html

[†] Informatics and Mathematical Modelling, Building 321, Technical University of Denmark, DK-2800 Lyngby, Denmark (pch@imm.dtu.dk).

[‡] Current address: TNM Consult, Marielundvej 48, DK-2730 Herlev, Denmark (toke.jensen@gmail.com).

Probably the newest member of the family of regularizing iteration methods is the GMRES algorithm [15]. If $A$ is symmetric, then GMRES is analytically identical to the MINRES algorithm [14], the latter yielding a simpler implementation with a short recursion. Regularizing GMRES and MINRES iterations were recently studied in [1], [2], [3], and [12].

The use of a matrix $L \neq I_n$ in the Tikhonov problem (1.1) can lead to better regularized solutions than the choice $L = I_n$, the explanation being that with a proper choice of $L$ the solution $x_\lambda$ is expressed in terms of basis vectors that are better suited to the problem. The choice of $L$ is problem dependent. As demonstrated by Hanke and Hansen [8], the matrix $L$ can be incorporated into the CGLS algorithm for solving $\min \|A\,x - b\|_2$ in such a way that the modified Krylov subspace provides the desired basis for the solution.

The purpose of this paper is to give a rigorous explanation of how we can carry this idea of preconditioning over to regularizing minimum-residual methods for a general smoothing norm $\|L\,\cdot\,\|_2$. The hope is that if the minimum-residual methods produce regularized solutions similar to the Tikhonov solutions, then incorporating the smoothing norm will produce solutions comparable to the general-form Tikhonov solutions. The main difficulty is that the smoothing-norm preconditioning from [8] typically involves rectangular matrices and therefore does not immediately carry over to the methods studied here. We shall demonstrate that we can still use the underlying idea, but the practical details and the implementation are different. Our preconditioner has the additional feature that it, when used in connection with symmetric problems, preserves the symmetry of the iteration matrix, thus allowing MINRES and MR-II to be used.

Since there is no overall "best" regularization algorithm, we believe that users should preferably have access to a variety of efficient and robust regularization methods. Also, a full understanding of the theoretical properties of regularizing iterations has not emerged and is a topic of current research. The goal of this paper is therefore not to emphasize preconditioned minimum-residual methods over other regularizing iterations, but instead to demonstrate how the preconditioner should be defined for a general matrix $L$ and implemented efficiently.

Our paper is organized as follows. Section 2 describes how to incorporate the matrix $L$ into regularizing CGLS iterations via a standard-form transformation based on the $A$-weighted pseudoinverse of $L$. In section 3 we briefly summarize a method based on augmentation of $L$ to a square matrix. Our main results are given in section 4, where we introduce our rectangular preconditioning technique, and in section 5 we demonstrate how to implement the new preconditioner efficiently. Finally, we illustrate our algorithm with one- and two-dimensional examples in section 6.

Throughout the paper, $I_q$ is the identity matrix of order $q$, $A^\dagger$ is the pseudoinverse of $A$, $\mathcal{R}(\cdot)$ and $\mathcal{N}(\cdot)$ denote the range and null space of a matrix, and the Krylov subspace is denoted by $\mathcal{K}_k(A, b) = \mathrm{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}$.

## 2. Working with smoothing norms.

We first summarize the results from [8] about smoothing norms. The key idea is to transform the general-form Tikhonov problem (1.1) into a problem in standard form,

$$\min_x \left\{ \|\bar{A}\,\bar{x} - \bar{b}\|_2^2 + \lambda^2\,\|\bar{x}\|_2^2 \right\}.$$

When $L$ is invertible, the standard-form transformation is easy: set $\bar{A} = A\,L^{-1}$ and $\bar{b} = b$, and use $\bar{x} = L\,x \Leftrightarrow x = L^{-1}\bar{x}$.

Often the matrix $L$ is rectangular and therefore not invertible. For example, if the smoothing norm $\|L\,x\|_2$ represents the norm of the first or second derivative of the solution, and if $x$ represents samples of the solution on a regular grid, then as $L$ we use the matrices $L_1 \in \mathbb{R}^{(n-1)\times n}$ and $L_2 \in \mathbb{R}^{(n-2)\times n}$ given by

$$L_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}, \qquad L_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix}. \tag{2.1}$$

With these rectangular matrices, the smoothing norm $\|L\,x\|_2$ is a seminorm. The matrices $L_1$ and $L_2$ are chosen such that their null spaces

$$\mathcal{N}(L_1) = \operatorname{span}\left\{(1,1,\dots,1)^T\right\}, \qquad \mathcal{N}(L_2) = \operatorname{span}\left\{(1,1,\dots,1)^T, (1,2,\dots,n)^T\right\}$$

represent the null spaces of the underlying first and second derivative operators. Obviously, any component of the solution in $\mathcal{N}(L)$ is unaffected by the regularization in (1.1), but since $\mathcal{N}(L_1)$ and $\mathcal{N}(L_2)$ are spanned by very smooth vectors (representing the constant and the linear functions), there is no harm in leaving the component unregularized.

To deal with such rectangular matrices, we assume that the matrix $L \in \mathbb{R}^{p\times n}$ satisfies $\operatorname{rank}(L) = p < n$. Then it is demonstrated in [8] that the standard-form transformation takes the form

$$\bar{A} = A\,L_A^\dagger, \qquad \bar{b} = b - A\,x_0, \qquad x_\lambda = L_A^\dagger \bar{x}_\lambda + x_0,$$

where $L_A^\dagger$ is the $A$-weighted pseudoinverse of $L$ (cf. [5]) given by

$$L_A^\dagger = E\,L^\dagger, \qquad \text{with} \quad E = I_n - \bigl(A\,(I_n - L^\dagger L)\bigr)^\dagger A, \tag{2.2}$$

and $x_0$ is the component of the solution lying in the null space of $L$,

$$x_0 = \bigl(A\,(I_n - L^\dagger L)\bigr)^\dagger\, b = N\,(A\,N)^\dagger\, b,$$

in which $N$ is any matrix of full column rank such that $\mathcal{R}(N) = \mathcal{N}(L)$.

To incorporate the smoothing norm into the framework of regularizing iterations, we apply CGLS to $\|\bar{A}\,\bar{x} - \bar{b}\|_2$, and Hanke and Hansen [8] demonstrated how the CGLS algorithm can be modified in such a way that all operations with $L_A^\dagger$ act as preconditioning. To see this, following section 6.1 of [10], we note that if $\mathcal{P}_k$ is the Ritz polynomial associated with $k$ steps of CG applied to $\bar{A}^T\bar{A}\,\bar{x} = \bar{A}^T\bar{b}$, then the iterate $x^{(k)}$ after $k$ steps of the preconditioned CGLS algorithm can be written as

$$x^{(k)} = \mathcal{P}_k\bigl(L_A^\dagger L_A^{\dagger T} A^T A\bigr)\, L_A^\dagger L_A^{\dagger T} A^T b + x_0. \tag{2.3}$$

It is now obvious that $L_A^\dagger L_A^{\dagger T}$ acts like a "preconditioner," and efficient methods for implementing this kind of preconditioning for CGLS and other methods are described in [8], [9], and [10, section 2.3.2]. We refer to the preconditioned version of CGLS as P-CGLS.

In some applications we encounter $L$ matrices with more rows than columns, typically in connection with Sobolev norms such as

$$\|L\,x\|_2^2 = \|L_1\,x\|_2^2 + \|L_2\,x\|_2^2 = \left\| \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} x \right\|_2^2 .$$

In this case an orthogonal factorization of $L$ can often lead to a more efficient implementation. Specifically, if $L = Q\,R$, where $Q$ has orthonormal columns and $R$ is triangular or trapezoidal and has full row rank, then $\|L\,x\|_2 = \|R\,x\|_2$, and we can thus replace $L$ with $R$. This approach can also be used in connection with P-CGLS because $L^\dagger = R^\dagger Q^T$ and $L^\dagger L = R^\dagger R$, and therefore $L_A^\dagger L_A^{\dagger T} = R_A^\dagger R_A^{\dagger T}$, showing that the underlying Krylov subspaces in (2.3) are identical.

Unfortunately, the preconditioner based on $\bar{A}$ cannot be applied to regularizing minimum-residual methods such as MINRES and GMRES because these methods require a square coefficient matrix, which is not the case for $\bar{A}$ when $L$ is noninvertible. Hence we need to develop a different kind of preconditioning for these methods.

**3. The augmented-matrix approach.** To be able to use GMRES/MINRES and the variants RRGMRES [1] and MR-II [6] (i.e., GMRES and MINRES with starting vector $Ab$ instead of $b$), the coefficient matrix must be square. When working with a rectangular $L$, such as in (2.1), it was suggested in [4] to augment it with additional rows to make it square and invertible. For $L_1$ and $L_2$, this approach leads to the augmented $n \times n$ matrices

$$\widehat{L}_1 = \begin{pmatrix} L_1 \\ w^T \end{pmatrix}, \qquad \widehat{L}_2 = \begin{pmatrix} \bar{w}^T \\ L_2 \\ w^T \end{pmatrix}. \tag{3.1}$$

If the additional rows are chosen such that the augmented matrices are invertible, then we can use the matrices $A\,\widehat{L}_1^{-1}$ and $A\,\widehat{L}_2^{-1}$ in connection with the minimum-residual methods. The use of a full-rank matrix $\widehat{L}$ in the standard-form transformation is equivalent to using $\widehat{L}^{-1}$ as a right preconditioner. We refer to [4] for more details about the choices of $w$ and $\bar{w}$.

While this augmented-matrix approach is simple to implement and use, it also has some disadvantages. For example, symmetry of the coefficient matrix $A$ does not carry over to the coefficient matrices $A\widehat{L}_1^{-1}$ and $A\widehat{L}_2^{-1}$, thus excluding the use of MINRES.[1] Moreover, any orthogonal reduction of $L$ changes the iterates because the Krylov subspace changes. For example, if $L$ is nonsingular and $L = QR$, then $\mathcal{K}_k(L^{-1}A, L^{-1}b) = \mathcal{K}_k(R^{-1}Q^TA, R^{-1}Q^Tb) \neq \mathcal{K}_k(R^{-1}A, R^{-1}b)$. This also rules out the use of any $L$ with more rows than columns.

**4. Smoothing-norm preconditioning.** As an alternative to the above technique, we now present an approach that works for any rectangular matrix $L \in \mathbb{R}^{p\times n}$ with $p < n$ and which does not require any modifications of the problem. In addition, our approach preserves symmetry, thus allowing short-recurrence implementations such as MINRES and MR-II to be used if $A$ is symmetric.

Our approach is similar in spirit to the technique described in section 2 for Tikhonov regularization and CGLS, but the details are different. We refer to the new preconditioned algorithms as SN-X, where X = GMRES, RRGMRES, MINRES, or MR-II, and SN is an abbreviation for "smoothing norm."

We start by writing the solution as the sum of the regularized component in $\mathcal{R}(L_A^\dagger)$ and the unregularized component in $\mathcal{N}(L)$,

$$x = L_A^\dagger y + x_0 = L_A^\dagger y + Nz, \tag{4.1}$$

---

[1] For symmetric $A$, one might instead consider applying MINRES to the system $\widehat{L}^{-T}A\widehat{L}^{-1}x = \widehat{L}^{-T}b$, where $\widehat{L}^{-T}A\widehat{L}^{-1}$ is symmetric.

where again $x_0 = N\,(A\,N)^\dagger\,b$ and $N$ is a matrix with full column rank whose columns span $\mathcal{N}(L)$. These columns need not be orthonormal, although this is preferable for numerical computations. The two vectors $y$ and $z = (A\,N)^\dagger b$ are uniquely determined because $L$ and $N$ both have full rank.

Our basic problem $A\,x = b$ can now be formulated as

$$A\,\left(L_A^\dagger\,,\,N\right)\begin{pmatrix} y \\ z \end{pmatrix} = b.$$

Premultiplication of this system with $\left(L_A^\dagger\,,\,N\right)^T$ leads to the $2\times 2$ block system

$$\begin{pmatrix} L_A^{\dagger T} A\,L_A^\dagger & L_A^{\dagger T} A\,N \\ N^T A\,L_A^\dagger & N^T A\,N \end{pmatrix}\begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} L_A^{\dagger T} b \\ N^T b \end{pmatrix}.$$

We eliminate $z$ from this system by forming the Schur complement system $S\,y = d$ with $S$ and $d$ given by

$$S = L_A^{\dagger T} A\,L_A^\dagger - L_A^{\dagger T} A\,N(N^T A\,N)^{-1} N^T A\,L_A^\dagger = L_A^{\dagger T} P\,A\,L_A^\dagger, \tag{4.2}$$

$$d = L_A^{\dagger T} b - L_A^{\dagger T} A\,N(N^T A\,N)^{-1} N^T b = L_A^{\dagger T} P\,b, \tag{4.3}$$

where we have introduced $P = I_n - AN(N^T AN)^{-1}N^T$. We shall now study the Schur system $S\,y = d$ in more detail.

THEOREM 4.1. [illegible] $\mathcal{R}(L^T)$ [illegible] $\mathcal{R}(AN)$ [illegible][2] [illegible]

$$P = I_n - AN(N^T AN)^{-1}N^T \tag{4.4}$$

[illegible] $\mathcal{R}(L^T)$ [illegible] $\mathcal{R}(AN)$

[illegible]. The matrix $I_n - P$ is idempotent because

$$(I_n - P)^2 = AN(N^T AN)^{-1}N^T AN(N^T AN)^{-1}N^T = I_n - P,$$

and hence it is a projector. Since $I_n - P$ is nonsymmetric, it is an oblique projector, and it is easy to see that the projection is onto $\mathcal{R}(AN)$ with $\mathcal{R}(L^T)$ contained in the null space. The assumption that $\mathcal{R}(L^T)$ and $\mathcal{R}(AN)$ are complementary subspaces ensures that $P$ is an oblique projector onto $\mathcal{R}(L^T)$ along $\mathcal{R}(AN)$. ☐

Smoothing-norm preconditioning for GMRES amounts to applying GMRES to the Schur complement system $Sy = d$. We emphasize that, as for CGLS, the purpose of this preconditioning is to provide a more desirable Krylov subspace for the regularized solution.

When we apply GMRES to the Schur system $Sy = d$, there exists a polynomial $\widetilde{\mathcal{P}}_k$ such that the solution after $k$ iterations is given by

$$y^{(k)} = \widetilde{\mathcal{P}}_k\left(L_A^{\dagger T} P A\,L_A^\dagger\right) L_A^{\dagger T} P\,b.$$

The corresponding vector $x^{(k)}$ is given by $x^{(k)} = L_A^\dagger y^{(k)} + x_0$, and we therefore obtain the SN-GMRES iterate

$$\begin{aligned} x^{(k)} &= L_A^\dagger \widetilde{\mathcal{P}}_k\left(L_A^{\dagger T} P A L_A^\dagger\right) L_A^{\dagger T} P\,b + x_0 \\ &= \widetilde{\mathcal{P}}_k\left(L_A^\dagger L_A^{\dagger T} P A\right) L_A^\dagger L_A^{\dagger T} P\,b + x_0, \end{aligned} \tag{4.5}$$

---

[2] The subspaces $\mathcal{R}(L^T) \subseteq \mathbb{R}^n$ and $\mathcal{R}(AN) \subseteq \mathbb{R}^n$ are complementary if $\mathcal{R}(L^T) + \mathcal{R}(AN) = \mathbb{R}^n$ and $\mathcal{R}(L^T) \cap \mathcal{R}(AN) = \{0\}$; see, e.g., [13, section 5.9].

showing that $x^{(k)} - x_0$ lies in the Krylov subspace $\mathcal{K}_k(L_A^\dagger L_A^{\dagger T} PA, L_A^\dagger L_A^{\dagger T} Pb)$. We note that the iterates of RRGMRES take the same form as for GMRES, except that the polynomial coefficients are different, and thus RRGMRES can also be used on the Schur system.

Although the polynomial expressions for the preconditioned CGLS and GMRES methods in (2.3) and (4.5) are similar in essence, the solutions obtained from the two methods are different, due to CGLS being a Ritz–Galerkin method and GMRES being a minimum-residual method. Even when $L$ is invertible, the two approaches produce different iterates. Furthermore, the oblique projector $P$ also indicates that the SN-X algorithms do not solve the same problem as does P-CGLS. Nevertheless, as we illustrate in section 6, our preconditioned algorithms are able to produce good solutions for certain problems.

We emphasize that the two main difficulties with the augmented-matrix approach from section 3 are both satisfactorily dealt with in this new approach. For a symmetric matrix $A$, the matrix $L_A^{\dagger T} P\, A\, L_A^\dagger$ is also symmetric, which follows from the symmetry of $PA = A - AN\,(N^TA\,N)^{-1}N^TA$. This symmetry allows us to use MINRES or MR-II on the Schur system, resulting in SN-MINRES and SN-MR-II. The new approach also allows us to use any rectangular $L$, including those with more rows than columns via the orthogonal reduction $L = QR$ mentioned in section 2.

**5. Implementation issues.** In this section we consider some issues that are important for the efficient implementation of the SN-X algorithms. We start with a theorem that simplifies the Schur system.

THEOREM 5.1. [illegible] 4.1 [illegible] $S\,y = d$ [illegible] (4.2)–(4.3) [illegible]

$$L^{\dagger T} P\, A\, L^\dagger y = L^{\dagger T} P\, b, \tag{5.1}$$

[illegible] $P$ [illegible] (4.4)

[illegible]. From the relation $(A(I_n - L^\dagger L))^\dagger = (ANN^\dagger)^\dagger = N(AN)^\dagger$ it follows that the matrix $E$ in (2.2) can be written as $E = I_n - N(AN)^\dagger A$. Moreover,

$$\begin{aligned} A^T(AN)^{\dagger T}N^TP &= A^T(AN)^{\dagger T}N^T - A^T(AN)^{\dagger T}N^TAN(N^TAN)^{-1}N^T \\ &= A^T(AN)^{\dagger T}N^T - A^T(AN)^{\dagger T}N^T = 0, \end{aligned}$$

and therefore $E^TP = (I_n - A^T(AN)^{\dagger T}N^T)P = P$. We also have the relation

$$\begin{aligned} P\,A\,N(AN)^\dagger A &= \big(I_n - AN(N^TAN)^{-1}N^T\big)AN(AN)^\dagger A \\ &= AN(AN)^\dagger A - AN(AN)^\dagger A = 0, \end{aligned}$$

and thus $PAE = PA(I_n - N(AN)^\dagger A) = PA$. Inserting these relations and $L_A^\dagger = EL^\dagger$ into the Schur system, we obtain (5.1). ☐

This theorem has an important impact on the numerical implementation of our preconditioner, because the weighted pseudoinverse $L_A^\dagger$ can be replaced by the ordinary pseudoinverse $L^\dagger$ in the computations. The weighted pseudoinverse $L_A^\dagger$ is needed only in the back-transformation (4.1).

Turning to the details of the implementation, we need to compute $x_0$ efficiently. This is done via the QR factorization of the matrix $AN$, which is always "skinny" for the low-dimensional null spaces associated with the derivative operators:

$$\begin{aligned} &1.\quad AN = Q_0R_0 \text{ (skinny QR factorization)}, \\ &2.\quad x_0 \leftarrow N\,R_0^{-1}Q_0^Tb. \end{aligned} \tag{5.2}$$

We also need an efficient technique for multiplications with $P$ from (4.4), which basically amounts to a number of "skinny" matrix-vector products. Using again the QR factorization of $AN$, we obtain

$$AN\,(N^TAN)^{-1}N^T = Q_0R_0(N^TQ_0R_0)^{-1}N^T = Q_0\,(N^TQ_0)^{-1}N^T,$$

and thus the product $P\,x$ is computed as

$$P\,x = x - Q_0\,(N^TQ_0)^{-1}N^Tx,$$

where a precomputed factorization of the small square matrix $N^TQ_0$ should be used. Assuming that $N$ has orthonormal columns, the smallest singular value of $N^TQ_0$ is equal to cosine of the subspace angle between $\mathcal{N}(L)$ and $\mathcal{R}(AN)$. Our experience is that this angle is usually small (because the smooth basis vectors of the two subspaces resemble each other), and consequently $N^TQ_0$ is well conditioned—however, there is no guarantee that this is always the case.

The complete algorithm for performing the multiplication $v = L^{\dagger T}PAL^{\dagger}y$ in the SN-X algorithms thus takes the form

$$\begin{aligned} &1.\quad v_1 \leftarrow A\,(L^{\dagger}y),\\ &2.\quad v_2 \leftarrow Q_0\,(N^TQ_0)^{-1}N^Tv_1,\\ &3.\quad v \leftarrow L^{\dagger T}(v_1 - v_2). \end{aligned} \tag{5.3}$$

The cost of working with the Schur complement system is therefore for each iteration the following: one multiplication with $A$, one with $L^{\dagger}$, one with $L^{\dagger T}$, and one with the oblique projector $P$. The preconditioning technique is feasible when the computation of $L^{\dagger}y$ and $L^{\dagger T}(v_1 - v_2)$ can be implemented efficiently, and the null space $\mathcal{N}(L)$ has low dimension such that multiplication with $P$ is inexpensive. The remaining work, i.e., reorthogonalization and updating of the residual norm and the solution, etc., is identical to applying the minimum-residual method to a nonpreconditioned system. A complete SN-X algorithm thus takes the following form:

(5.4)
1. Use (5.2) to compute $Q_0$, $R_0$, and $x_0$.
2. Run $k$ steps of algorithm X on (5.1), using (5.3), to compute $y^{(k)}$.
3. Set $x^{(k)} = L_A^{\dagger}y^{(k)} + x_0$.

While not fully documented in [8], [9], [10], P-CGLS—in addition to the multiplications with $A$ and $A^T$—also requires multiplications with $L^{\dagger}$ and its transpose, as well as one multiplication with the matrix $E$ (which is also an oblique projector). Hence the overall work for preconditioning the SN-X algorithms is essentially the same as that for P-CGLS.

We use an example from two-dimensional problems to illustrate how to work with a rank-deficient $L$ with more rows than columns. Assume that $X \in \mathbb{R}^{M\times N}$ is the two-dimensional solution, and that we use the Sobolev norm $\|L_{d_2}X\|_{\mathrm{F}}^2 + \|X\,L_{d_1}^T\|_{\mathrm{F}}^2$, where $L_{d_1} \in \mathbb{R}^{(N-d_1)\times N}$ and $L_{d_2} \in \mathbb{R}^{(M-d_2)\times M}$ are the matrices in (2.1). Then $L$ takes the form

$$L = \begin{pmatrix} L_{d_1} \otimes I_M \\ I_N \otimes L_{d_2} \end{pmatrix}, \tag{5.5}$$

and we note that $L$ has more rows than columns and is rank-deficient. The following theorem shows how to proceed via the SVDs of the "small" matrices $L_{d_1}$ and $L_{d_2}$.

Theorem 5.2. [illegible] $L$ [illegible] (5.5) [illegible] $L_{d_1} = U_{d_1}\,\Sigma_{d_1}\,V_{d_1}^T$ [illegible] $L_{d_2} = U_{d_2}\,\Sigma_{d_2}\,V_{d_2}^T$ [illegible] $L_{d_1}$ [illegible] $L_{d_2}$ [illegible] $\|L\,x\|_2 = \|L_D\,x\|_2$ [illegible]

$$L_D = D\,(V_{d_1} \otimes V_{d_2})^T, \tag{5.6}$$

[illegible] $D \in \mathbb{R}^{MN\times MN}$ [illegible]

$$D^2 = \Sigma_{d_1}^T\Sigma_{d_1} \otimes I_M + I_N \otimes \Sigma_{d_2}^T\Sigma_{d_2}. \tag{5.7}$$

[illegible] $V_{d_i} = (\,\overline{V}_{d_i}\,,\,N_{d_i}\,)$ [illegible] $\mathcal{N}(L_{d_i}) = \mathcal{R}(N_{d_i})$ [illegible] $i = 1, 2$ [illegible] $\mathcal{N}(L) = \mathcal{N}(L_D)$ [illegible]

$$N = N_{d_1} \otimes N_{d_2}. \tag{5.8}$$

[illegible]. Inserting the SVDs of $L_{d_1}$ and $L_{d_2}$ and using $I_N = V_{d_1}V_{d_1}^T$ and $I_M = V_{d_2}V_{d_2}^T$, we obtain

$$\begin{aligned} L &= \begin{pmatrix} L_{d_1} \otimes I_M \\ I_N \otimes L_{d_2} \end{pmatrix} = \begin{pmatrix} (U_{d_1}\,\Sigma_{d_1}\,V_{d_1}^T) \otimes (V_{d_2}\,V_{d_2}^T) \\ (V_{d_1}\,V_{d_1}^T) \otimes (U_{d_2}\,\Sigma_{d_2}\,V_{d_2}^T) \end{pmatrix} \\ &= \begin{pmatrix} U_{d_1} \otimes V_{d_2} & 0 \\ 0 & V_{d_1} \otimes U_{d_2} \end{pmatrix} \begin{pmatrix} \Sigma_{d_1} \otimes I_M \\ I_N \otimes \Sigma_{d_2} \end{pmatrix} (V_{d_1} \otimes V_{d_2})^T. \end{aligned}$$

Since the middle matrix consists of two "stacked" diagonal matrices, we can easily determine an orthogonal matrix $Q_D$ (consisting of Givens rotations) and a diagonal matrix $D$ such that

$$Q_D^T \begin{pmatrix} \Sigma_{d_1} \otimes I_M \\ I_N \otimes \Sigma_{d_2} \end{pmatrix} = \begin{pmatrix} D \\ 0 \end{pmatrix},$$

and it is no restriction to assume that the diagonal elements of $D$ are nonnegative. Hence

$$L = \begin{pmatrix} U_{d_1} \otimes V_{d_2} & 0 \\ 0 & V_{d_1} \otimes U_{d_2} \end{pmatrix} Q_D \begin{pmatrix} D \\ 0 \end{pmatrix} (V_{d_1} \otimes V_{d_2})^T,$$

and we obtain $\|L\,x\|_2 = \|L_D\,x\|_2$. Equation (5.7) follows from the relation

$$D^2 = D^TD = \begin{pmatrix} D \\ 0 \end{pmatrix}^T \begin{pmatrix} D \\ 0 \end{pmatrix} = \begin{pmatrix} \Sigma_{d_1} \otimes I_M \\ I_N \otimes \Sigma_{d_2} \end{pmatrix}^T \begin{pmatrix} \Sigma_{d_1} \otimes I_M \\ I_N \otimes \Sigma_{d_2} \end{pmatrix}.$$

Regarding the null space $\mathcal{N}(L) = \mathcal{N}(L_D)$, it is easily seen from (5.6) that the null space vectors are given by the columns of $V_{d_1} \otimes V_{d_2}$ for which the diagonal elements of $D$ are zero. This leads directly to (5.8). □

The consequences of this theorem are that we can substitute the structured matrix $L_D$ for $L$ in the Tikhonov problem (1.1), and that we have a simple basis for $\mathcal{N}(L)$. The approach can be used in both the P-CGLS and the SN-X algorithms and leads to increased efficiency, while $L$ matrices of this form are [illegible] applicable in the augmented-matrix approach.

**6. Numerical experiments.** We include two test problems to illustrate the performance of the new preconditioning technique for regularizing minimum-residual methods. The first example is a synthetic two-dimensional problem with a symmetric coefficient matrix and an $L$ of the form (5.5); the second is a finite-element model of a steady-state heat distribution problem with a nonsymmetric coefficient matrix and a periodic solution. In both examples we calculate the relative error of the regularized solutions $x^{(k)}$ compared to the exact solution $x$, i.e.,

$$\epsilon^{(k)} = \|x^{(k)} - x\|_2 \,/\, \|x\|_2, \tag{6.1}$$

where $x^{(k)}$ is the $k$th iterate. The best regularized solution is always defined as the solution for which $\epsilon^{(k)}$ is smallest.

**6.1. Two-dimensional example.** The two-dimensional example is a synthetic problem generated from the `deriv2` test problem in Regularization Tools [9]. The coefficient matrix $A \in \mathbb{R}^{37500\times 37500}$ is generated as the Kronecker product of $A_1 \in \mathbb{R}^{250\times 250}$ and $A_2 \in \mathbb{R}^{150\times 150}$, each being a discretization of Green's function for the second derivative. The resulting coefficient matrix is symmetric, and thus MINRES and MR-II can be used.

The exact solution $X$ has size $M \times N = 150 \times 250$, and it consists of the sum of a linear function in one direction and a quadratic function in the other. The MATLAB code for generating the exact solution is

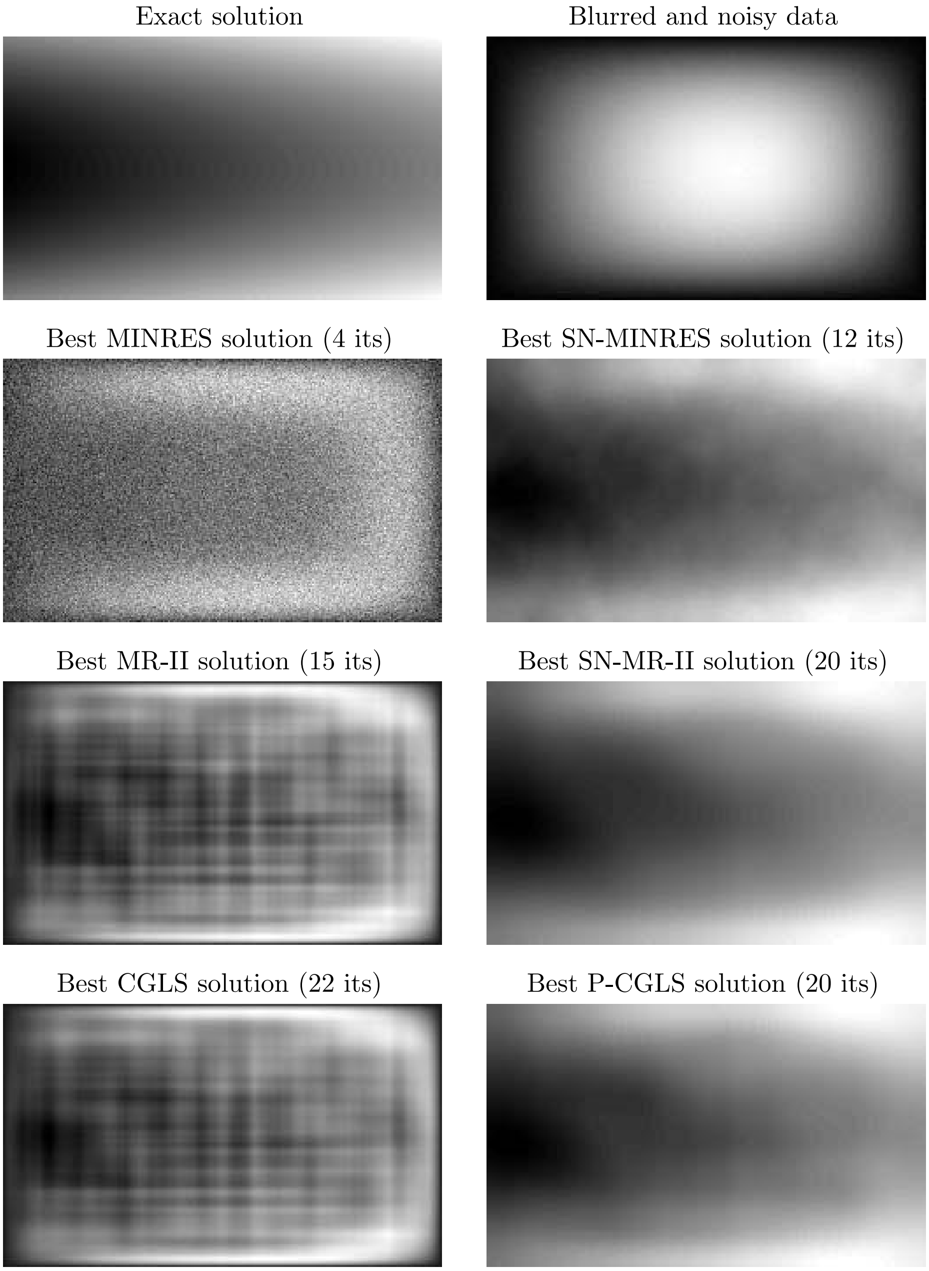
```
s = linspace(-1,1,250); t = linspace(-1,1,150);
[s,t] = meshgrid(s,t);
X = s + t.^2;
```

and then $x$ is obtained by stacking the columns of $X$. The right-hand side is computed by $b = Ax+e$, in which $e$ is white Gaussian noise scaled such that $\|e\|_2/\|Ax\|_2 = 10^{-2}$. Using the Kronecker products, all computations can be carried out without explicitly forming the large matrix $A$, instead using $A_1$ and $A_2$.

The top plots in Figure 6.1 show the exact solution and the right-hand side as two-dimensional images. Figure 6.1 also shows the best regularized solutions using MINRES, MR-II, CGLS, SN-MINRES, SN-MR-II, and P-CGLS as measured by (6.1). For the preconditioned algorithms we use the matrix $L$ in (5.5) with $d_1 = d_2 = 1$, corresponding to first derivative smoothing in both directions, and its null space is spanned by the constant image.

The six algorithms produce solutions with different properties; all the preconditioned algorithms give much better regularized solutions than the nonpreconditioned algorithms, and MINRES gives by far the most noisy solution. All the nonpreconditioned solutions tend to go to zero near the edges of the domain; this behavior comes from the fact that the bases for these solutions are the Krylov vectors, which all tend to zero at the edges. Using the matrix $L$ from (5.5) as preconditioner, the Krylov subspaces are changed to favor the particular properties of this problem, resulting in better reconstructions. The best SN-MR-II solution is similar to the best solution obtained by P-CGLS, both regarding the quality and the number of iterations. It is worth noting that even though the cost of applying the preconditioner is about the same for P-CGLS and SN-MR-II, P-CGLS needs an additional matrix-vector multiplication with $A^T$ in each iteration, which makes SN-MR-II somewhat faster.

**6.2. Steady-state heat distribution.** This test problem from [11] involves a partial differential equation that describes the steady-state heat distribution in a two-dimensional domain $\Omega$ with inner and outer boundaries $\partial\Omega_{\rm i}$ and $\partial\Omega_{\rm o}$. The forward



Exact solution

Blurred and noisy data

Best MINRES solution (4 its)

Best SN-MINRES solution (12 its)

Best MR-II solution (15 its)

Best SN-MR-II solution (20 its)

Best CGLS solution (22 its)

Best P-CGLS solution (20 its)

FIG. 6.1. *Top row: exact solution and right-hand side for the two-dimensional test problem with a symmetric coefficient matrix. The remaining rows show the best regularized solutions using standard and preconditioned algorithms. The preconditioning uses the matrix* $L$ *in* (5.5) *with* $d_1 = d_2 = 1$ *corresponding to first derivative smoothing in both the vertical and horizontal directions.*

problem takes the form

$$\nabla^2 f(z) = 0, \quad z \in \Omega, \qquad \qquad \begin{cases} f(z) = f_{\mathrm{i}}(z), & z \in \partial\Omega_{\mathrm{i}}, \\ \frac{\partial}{\partial n} f(z) = 0, & z \in \partial\Omega_{\mathrm{o}}, \end{cases}$$

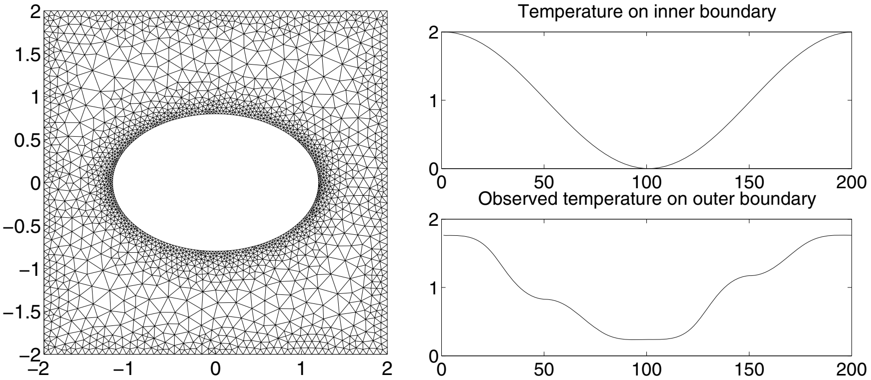


FIG. 6.2. *Test problem with steady-state heat distribution. Left: the geometry and the finite-element mesh. Right: exact temperature $f_i$ on the inner boundary (top), and measured noisy temperature $f_o$ on the outer boundary (bottom).*

where $f_{\mathrm{i}}$ is the temperature on the inner boundary and $\frac{\partial}{\partial n}$ denotes the normal derivative on the outer boundary. The inverse problem is to determine the temperature distribution on the inner boundary from measurements of the temperature on the outer boundary.

In this example, the inner boundary $\partial\Omega_{\mathrm{i}}$ is an ellipse with semiaxes of length 1.2 and 0.8, while the outer boundary $\partial\Omega_{\mathrm{o}}$ is a square with side length 4; see Figure 6.2. We use a matrix-free implementation in which $A$, the forward computation, is a finite-element model that maps the temperature $f_{\mathrm{i}}(z)$ on the inner boundary $\partial\Omega_{\mathrm{i}}$ to the temperature $f_{\mathrm{o}}(z)$ on the outer boundary $\partial\Omega_{\mathrm{o}}$.

The exact solution vector $x$ consists of values of the temperature on the inner boundary in $n = 200$ grid points. The right-hand side, consisting of the temperature in the $n$ grid points of the outer boundary, is then computed as $b = Ax + e$, where $e$ is white Gaussian noise scaled such that $\|e\|_2/\|Ax\|_2 = 10^{-3}$. Both temperature profiles are also shown in Figure 6.2.

Using $n$ points on both $\partial\Omega_i$ and $\partial\Omega_o$, the operation with the square matrix $A$ involves solving the forward problem via the finite-element model. A similar discrete model for operation with the transposed matrix is not simple to derive, and therefore P-CGLS is not directly applicable. On the other hand, GMRES and RRGMRES are natural methods of choice for solving the inverse problem.

The two top rows in Figure 6.3 show the optimal solutions obtained with GMRES and RRGMRES using $L = I_n$ and $L = L_2$. The third row shows the comparable results with $L$ equal to the augmented matrix $\widehat{L}_2$ from (3.1) with $\bar{w} = \alpha(-2, 1, 0, \ldots, 0)^T$, $w = \alpha(0, \ldots, 0, 1, -2)^T$, and $\alpha = 4.29 \cdot 10^{-4}$. The $\alpha$-parameter can be considered as a special regularization parameter for the added rows and must be chosen appropriately in addition to the number of iterations. Selecting the optimal $\alpha$ is in general not an easy problem. For this constructed test problem we compute the optimal solutions for a range of $\alpha$ values, and the chosen $\alpha$ is the one that gives rise to the overall best regularized solution. Figure 6.4 shows how the optimal GMRES solutions vary with $\alpha$—the situation is very similar for RRGMRES.

For all three choices of $L$, RRGMRES performs better than GMRES. Using a smoothing norm with $L = L_2$ or $L = \widehat{L}_2$ improves the solutions by about an order of magnitude. With these two choices of $L$, the augmented-matrix approach seems to

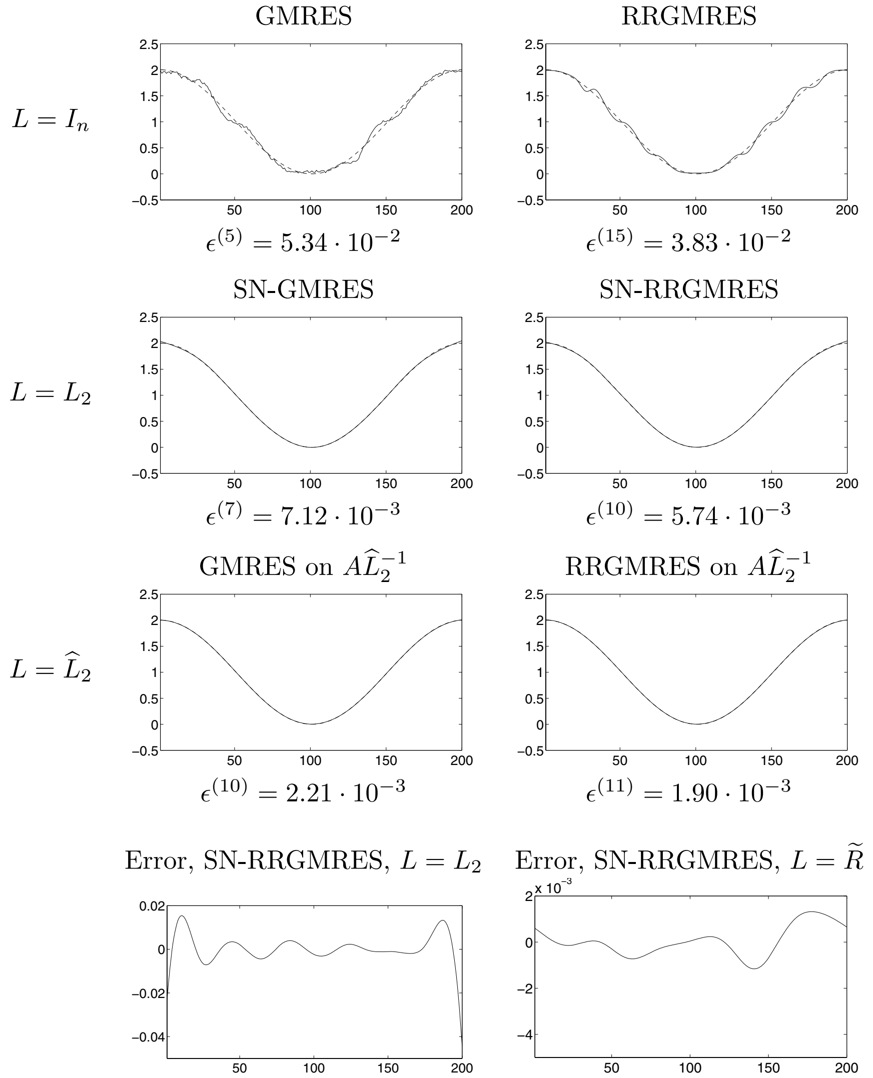


FIG. 6.3. *Three top rows: solutions to the steady-state heat distribution problem computed with standard and preconditioned GMRES and RRGMRES, and with different smoothing norms. The numbers $\epsilon^{(k)}$ below the plots are the optimal relative errors after $k$ iterations. Bottom row: the error $x - x^k$ in the SN-RRGMRES solution for two choices of $L$.*

perform better than the SN-approach. On the other hand, Figure 6.4 shows that a small change in $\alpha$ will result in worse solutions. An obvious choice of $\alpha$ would be to choose a value so small that the influence of the added rows is negligible while still keeping $\widehat{L}_2$ invertible, e.g., $\alpha = 10^{-8}$. This corresponds to the flat part of the plot in Figure 6.4, and an optimal solution with a higher relative error. The dependence on $\alpha$ illustrates the importance of choosing wise extensions to the $L$-matrices. Especially, the extension proposed in [4, Equation (13)] ($\widehat{L}_2$ with $\alpha = 1$) is seen to perform badly for this particular problem.

However, the quality of the solution can be improved further by taking into account the periodicity of the solution. A careful study shows that the errors in all the

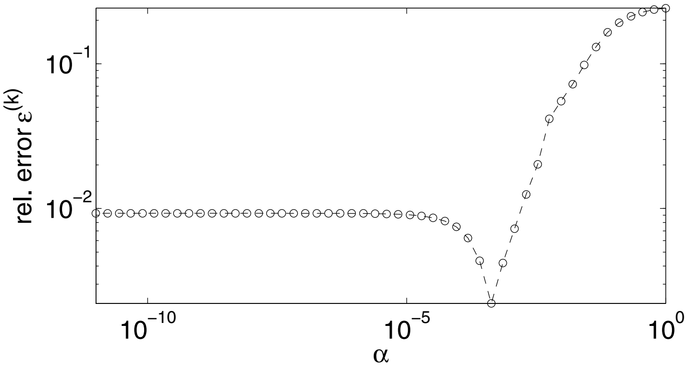


FIG. 6.4. *Best obtainable relative errors* $\epsilon^{(k)}$ *of GMRES solutions using* $\widehat{L}_2$ *with* $\bar{w} = \alpha(-2,\,1,\,0,\ldots,\,0)^T$ *and* $w = \alpha(0,\ldots,\,0,\,1,\,-2)^T$ *for a range of values of* $\alpha$.

solutions grow towards the edges; this is illustrated by the bottom left plot in Figure 6.3, which shows the error in the SN-RRGMRES solution with $L = L_2$. A natural requirement is therefore to add the constraint that the second derivative should also be minimized across the edges of the periodic solution. Hence, we use a circulant version of $L_2$ given by

$$\widetilde{L}_2 = \begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{n\times n},$$

which corresponds to $L_2$ with periodic boundary conditions. This matrix is singular with $\operatorname{rank}(\widetilde{L}_2) = n-1$, and its null space is given by

$$\mathcal{N}(\widetilde{L}_2) = \operatorname{span}\left\{(1,\,1,\,\ldots,\,1)^T\right\}.$$

Since $\widetilde{L}_2$ is singular, it cannot be used in the augmented-matrix approach. In order to use the SN-approach, we compute the skinny QR-factorization $\widetilde{L}_2 = \widetilde{Q}\widetilde{R}$ with $\widetilde{R}$ trapezoidal, and use $L = \widetilde{R}$.

Using this choice of $L$, the smallest error $\epsilon^{(6)} = 6.28 \cdot 10^{-4}$ in the SN-GMRES solution is obtained after 6 iterations, while the smallest error $\epsilon^{(5)} = 5.21 \cdot 10^{-4}$ in the SN-RRGMRES solution is obtained after 5 iterations. In both cases, the relative error is reduced by an order of magnitude, compared to using $L_2$, and is also superior to the approach using the augmented $\widehat{L}_2$ with $\alpha = 4.29 \cdot 10^{-4}$. Furthermore, the number of iterations is reduced. The bottom right plot in Figure 6.3, which shows the error in the SN-RRGMRES solution with $L = \widetilde{R}$, illustrates the reduction of the error, primarily because the errors near the edges are reduced significantly.

## 7. Conclusion.

We presented a new preconditioning technique for regularizing minimum-residual methods such that it corresponds to using a smoothing norm $\|L\,x\|_2$ in the Tikhonov formulation, and the matrix $L$ is allowed to be both rank deficient and rectangular. Our algorithm preserves symmetry when the coefficient matrix is symmetric, thus allowing the use of MINRES and MR-II where appropriate. Our algorithm is computationally feasible when the dimension of $\mathcal{N}(L)$ is small and computations with $L^\dagger$ can be implemented efficiently. We showed an example of an $L$ matrix for two-dimensional problems where this is achieved. We also demonstrated how to implement the algorithm efficiently, and we gave numerical examples in one and two dimensions that illustrate the use and performance of the new preconditioner.

Our numerical examples illustrate that the proposed SN-approach can provide an improvement in the solution's accuracy compared to standard minimum-residual methods, that the SN-approach works for a larger class of smoothing norms than the augmented-matrix approach, and that preconditioned minimum-residual methods can be computationally attractive alternatives to preconditioned CGLS (e.g., when $A^T$ is not directly available).

**Acknowledgements.** We would like to thank the referees for comments, which considerably improved the presentation and the numerical examples.

## REFERENCES

[1] D. Calvetti, B. Lewis, and L. Reichel, *GMRES-type methods for inconsistent systems*, Linear Algebra Appl., 316 (2000), pp. 157–169.

[2] D. Calvetti, B. Lewis, and L. Reichel, *GMRES, L-curves, and discrete ill-posed problems*, BIT, 42 (2002), pp. 44–65.

[3] D. Calvetti, B. Lewis, and L. Reichel, *On the regularizing properties of the GMRES method*, Numer. Math., 91 (2002), pp. 605–625.

[4] D. Calvetti, L. Reichel, and A. Shuibi, *Invertible smoothing preconditioners for linear discrete ill-posed problems*, Appl. Numer. Math., 54 (2005), pp. 135–149.

[5] L. Eldén, *A weighted pseudoinverse, generalized singular values, and constrained least squares problems*, BIT, 22 (1982), pp. 487–501.

[6] M. Hanke, *Conjugate Gradient Type Methods for Ill-Posed Problems*, Longman, Harlow, UK, 1995.

[7] M. Hanke, *On Lanczos based methods for the regularization of discrete ill-posed problems*, BIT, 41 (2001), pp. 1008–1018.

[8] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Industry, 3 (1993), pp. 253–315.

[9] P. C. Hansen, *Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

[10] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM Monogr. Math. Model. Comput. 4, SIAM, Philadelphia, 1997.

[11] M. Jacobsen, *Modular Regularization Algorithms*, Ph.D. thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 2004.

[12] M. Kilmer and G. W. Stewart, *Iterative regularization and MINRES*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 613–628.

[13] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[14] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal, 12 (1975), pp. 617–629.

[15] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 15–32

# MAJORIZATION FOR CHANGES IN ANGLES BETWEEN SUBSPACES, RITZ VALUES, AND GRAPH LAPLACIAN SPECTRA*

ANDREW V. KNYAZEV[†] AND MERICO E. ARGENTATI[†]

**Abstract.** Many inequality relations between real vector quantities can be succinctly expressed as "weak (sub)majorization" relations using the symbol $\prec_w$. We explain these ideas and apply them in several areas, angles between subspaces, Ritz values, and graph Laplacian spectra, which we show are all surprisingly related. Let $\Theta(\mathcal{X},\mathcal{Y})$ be the vector of principal angles in nondecreasing order between subspaces $\mathcal{X}$ and $\mathcal{Y}$ of a finite dimensional space $\mathcal{H}$ with a scalar product. We consider the change in principal angles between subspaces $\mathcal{X}$ and $\mathcal{Z}$, where we let $\mathcal{X}$ be perturbed to give $\mathcal{Y}$. We measure the change using weak majorization. We prove that $|\cos^2\Theta(\mathcal{X},\mathcal{Z})-\cos^2\Theta(\mathcal{Y},\mathcal{Z})|\prec_w \sin\Theta(\mathcal{X},\mathcal{Y})$, and give similar results for differences of cosines, i.e., $|\cos\Theta(\mathcal{X},\mathcal{Z})-\cos\Theta(\mathcal{Y},\mathcal{Z})|\prec_w \sin\Theta(\mathcal{X},\mathcal{Y})$, and of sines and sines squared, assuming $\dim\mathcal{X}=\dim\mathcal{Y}$. We observe that $\cos^2\Theta(\mathcal{X},\mathcal{Z})$ can be interpreted as a vector of Ritz values, where the Rayleigh–Ritz method is applied to the orthogonal projector on $\mathcal{Z}$ using $\mathcal{X}$ as a trial subspace. Thus, our result for the squares of cosines can be viewed as a bound on the change in the Ritz values of an orthogonal projector. We then extend it to prove a general result for Ritz values for an arbitrary Hermitian operator $A$, not necessarily a projector: let $\Lambda\left(P_{\mathcal{X}}A|_{\mathcal{X}}\right)$ be the vector of Ritz values in nonincreasing order for $A$ on a trial subspace $\mathcal{X}$, which is perturbed to give another trial subspace $\mathcal{Y}$; then $|\Lambda\left(P_{\mathcal{X}}A|_{\mathcal{X}}\right)-\Lambda\left(P_{\mathcal{Y}}A|_{\mathcal{Y}}\right)|\prec_w(\lambda_{\max}-\lambda_{\min})\,\sin\Theta(\mathcal{X},\mathcal{Y})$, where the constant is the difference between the largest and the smallest eigenvalues of $A$. This establishes our conjecture that the root two factor in our earlier estimate may be eliminated. Our present proof is based on a classical but rarely used technique of extending a Hermitian operator in $\mathcal{H}$ to an orthogonal projector in the "double" space $\mathcal{H}^2$. An application of our Ritz values weak majorization result for Laplacian graph spectra comparison is suggested, based on the possibility of interpreting eigenvalues of the edge Laplacian of a given graph as Ritz values of the edge Laplacian of the complete graph. We prove that $\sum_k|\lambda_k^1-\lambda_k^2|\le nl$, where $\lambda_k^1$ and $\lambda_k^2$ are all ordered elements of the Laplacian spectra of two graphs with the same $n$ vertices and with $l$ equal to the number of differing edges.

**Key words.** majorization, principal angles, canonical angles, canonical correlations, subspace, orthogonal projection, perturbation analysis, Ritz values, Rayleigh–Ritz method, graph spectrum, graph vertex Laplacian, graph edge Laplacian

**AMS subject classifications.** 15A42, 15A60, 65F35, 05C50

**DOI.** 10.1137/060649070

## 1. Introduction.

Many inequality relations between real vector quantities can be succinctly expressed as "weak (sub)majorization" relations using the symbol $\prec_w$ that we now introduce. For a real vector $x=[x_1,\dots,x_n]$ let $x^{\downarrow}$ be the vector obtained by rearranging the entries of $x$ in an algebraically nonincreasing order. Vector $y$ weakly majorizes vector $x$; i.e., $x\prec_w y$ if $\sum_{i=1}^k x_i^{\downarrow}\le\sum_{i=1}^k y_i^{\downarrow},\ k=1,\dots,n$. The importance of weak majorization can be seen from the classical statement that the following two conditions are equivalent: $x\prec_w y$ and $\sum_{i=1}^n\phi(x_i)\le\sum_{i=1}^n\phi(y_i)$ for all nondecreasing convex functions $\phi$. Thus, a single weak majorization result implies a

---

*Received by the editors January 4, 2006; accepted for publication (in revised form) by I. C. F. Ipsen March 31, 2006; published electronically December 21, 2006. Preliminary results of this paper were presented at the 12th ILAS Conference in Canada in June, 2005, at the minisymposium "Canonical Angles Between Subspaces: Theory and Applications." This material is based upon work supported by the National Science Foundation award DMS 0208773 and the Intelligence Technology Innovation Center through the joint "Approaches to Combat Terrorism" NSF Program Solicitation NSF 03-569.

http://www.siam.org/journals/simax/29-1/64907.html

[†]Department of Mathematical Sciences, University of Colorado at Denver, and Health Sciences Center, P.O. Box 173364, Campus Box 170, Denver, CO 80217-3364 (andrew.knyazev@cudenver.edu, http://math.cudenver.edu/∼aknyazev/, merico.argentati@cudenver.edu).

great variety of inequalities. We explain these ideas and apply them in several areas, angles between subspaces, Ritz values, and graph Laplacian spectra, which we show are all surprisingly related.

The concept of principal angles, also referred to as canonical angles, between subspaces is one of the classical mathematical ideas originating from [16] with many applications. In functional analysis, the gap between subspaces, which is related to the sine of the largest principal angle, bounds the perturbation of a closed linear operator by measuring the change in its graph, while the smallest nontrivial principal angle between two subspaces determines whether the sum of the subspaces is closed. In numerical analysis, principal angles appear naturally to estimate how close an approximate eigenspace is to the true eigenspace. The chordal distance, the Frobenius norm of the sine of the principal angles, on the Grassmannian space of finite dimensional subspaces is used, e.g., for subspace packing with applications in control theory. In statistics, the cosines of principal angles are called canonical correlations and have applications in information retrieval and data visualization.

Let $\mathcal{H}$ be a real or complex $n < \infty$ dimensional vector space equipped with an inner product $(x, y)$ and a vector norm $\|x\| = (x, x)^{1/2}$. The acute angle between two nonzero vectors $x$ and $y$ is defined as

$$\theta(x, y) = \arccos \frac{|(x, y)|}{\|x\|\|y\|} \in \left[0, \frac{\pi}{2}\right].$$

For three nonzero vectors $x, y, z$, we have bounds on the change in the angle

$$|\theta(x, z) - \theta(y, z)| \leq \theta(x, y), \tag{1.1}$$

in the sine

$$|\sin(\theta(x, z)) - \sin(\theta(y, z))| \leq \sin(\theta(x, y)), \tag{1.2}$$

and in the cosine

$$|\cos(\theta(x, z)) - \cos(\theta(y, z))| \leq \sin(\theta(x, y)), \tag{1.3}$$

and a more subtle bound on the change in the sine or cosine squared:

$$\left|\cos^2(\theta(x, z)) - \cos^2(\theta(y, z))\right| = \left|\sin^2(\theta(x, z)) - \sin^2(\theta(y, z))\right| \leq \sin(\theta(x, y)). \tag{1.4}$$

Let us note that we can project the space $\mathcal{H}$ into the span$\{x, y, z\}$ without changing the angles; i.e., the inequalities above present essentially the case of a three dimensional (3D) space.

Inequality (1.1) is proved in [34, Theorem 3.2, p. 514]. We note that (1.2) follows from (1.1), since the sine function is increasing and subadditive; see [34, p. 530].

It is instructive to provide a simple proof of the sine inequality (1.2) using orthogonal projectors. Let $P_{\mathcal{X}}$, $P_{\mathcal{Y}}$, and $P_{\mathcal{Z}}$ be the orthogonal projectors onto the subspaces spanned by the vectors $x$, $y$, and $z$, respectively, and let $\|\cdot\|$ also denote the induced operator norm. When we are dealing with 1D subspaces, we have the following elementary formula: $\sin(\theta(x, y)) = \|P_{\mathcal{X}} - P_{\mathcal{Y}}\|$. (Indeed, $P_x - P_y$ has rank at most two, so it has at most two nonzero singular values, but $(P_x - P_y)^2 x = (1 - |(x, y)|^2)x$ and $(P_x - P_y)^2 y = (1 - |(x, y)|^2)y$ for unit vectors $x$ and $y$, so $1 - |(x, y)|^2 = \sin^2(\theta(x, y))$ is a double eigenvalue of $(P_x - P_y)^2$.) Then the sine inequality (1.2) is equivalent to the triangle inequality $|\,\|P_{\mathcal{X}} - P_{\mathcal{Z}}\| - \|P_{\mathcal{Y}} - P_{\mathcal{Z}}\|\,| \leq \|P_{\mathcal{X}} - P_{\mathcal{Y}}\|$.

In this paper, we replace 1D subspaces spanned by the vectors $x$, $y$, and $z$ with multidimensional subspaces $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, and we use the concept of principal angles between subspaces. Principal angles are very well studied in the literature; however, some important gaps still remain. Here, we are interested in generalizing inequalities (1.2)–(1.4) above to multidimensional subspaces to include all principal angles, using weak majorization.

Let us denote by $\Theta(\mathcal{X},\mathcal{Y})$ the vector of principal angles in nondecreasing order between subspaces $\mathcal{X}$ and $\mathcal{Y}$. Let $\dim\mathcal{X} = \dim\mathcal{Y}$, and let another subspace $\mathcal{Z}$ be given. We prove that $|\cos^2\Theta(\mathcal{X},\mathcal{Z}) - \cos^2\Theta(\mathcal{Y},\mathcal{Z})| \prec_w \sin\Theta(\mathcal{X},\mathcal{Y})$, and give similar results for differences of cosines, i.e., $|\cos\Theta(\mathcal{X},\mathcal{Z}) - \cos\Theta(\mathcal{Y},\mathcal{Z})| \prec_w \sin\Theta(\mathcal{X},\mathcal{Y})$, and of sines and sines squared. This is the first main result of the present paper; see section 3. The proof of weak majorization for sines is a direct generalization of the 1D proof above. Our proofs of weak majorization for cosines and sines or cosines squared do not have such simple 1D analogues.

Pioneering results using angles between subspaces in the framework of unitarily invariant norms and symmetric gauge functions, equivalent to majorization, appear in [7], which introduces many of the tools that we use here. The main goal of [7] is, however, different—analyzing the perturbations of eigenvalues and eigenspaces—while in the present paper we are concerned with sensitivity of angles and Ritz values with respect to changes in subspaces.

Our second main result (see section 4) bounds the change in the Ritz values with the change of the trial subspace. We attack the problem by discovering a simple, but deep, connection between the principal angles and the Rayleigh–Ritz method.

We first give a brief definition of Ritz values. Let $A : \mathcal{H} \to \mathcal{H}$ be a Hermitian operator, and let $\mathcal{X}$ be a (so-called trial) subspace of $\mathcal{H}$. We define an operator $P_{\mathcal{X}}A|_{\mathcal{X}}$ on $\mathcal{X}$, where $P_{\mathcal{X}}$ is the orthogonal projector onto $\mathcal{X}$ and $P_{\mathcal{X}}A|_{\mathcal{X}}$ denotes the restriction of operator $P_{\mathcal{X}}A$ to its invariant subspace $\mathcal{X}$, as discussed, e.g., in [32, section 11.4, pp. 234–239]. The eigenvalues $\Lambda(P_{\mathcal{X}}A|_{\mathcal{X}})$ are called Ritz values of the operator $A$ with respect to the trial subspace $\mathcal{X}$.

We observe that the cosines squared $\cos^2\Theta(\mathcal{X},\mathcal{Z})$ of principal angles between subspaces $\mathcal{X}$ and $\mathcal{Z}$ can be interpreted as a vector of Ritz values, where the Rayleigh–Ritz method is applied to the orthogonal projector $P_{\mathcal{Z}}$ onto $\mathcal{Z}$ using $\mathcal{X}$ as a trial subspace. Let us illustrate this connection for one dimensional $\mathcal{X} = \operatorname{span}\{x\}$ and $\mathcal{Z} = \operatorname{span}\{z\}$, where it becomes trivial:

$$\cos^2(\theta(x,z)) = \frac{(x, P_{\mathcal{Z}}x)}{(x,x)}.$$

The ratio on the right is the Rayleigh quotient for $P_{\mathcal{Z}}$—the one dimensional analogue of a Ritz value. In this notation, estimate (1.4) turns into

$$\left|\frac{(x,P_z x)}{(x,x)} - \frac{(y,P_z y)}{(y,y)}\right| \leq \sin(\theta(x,y)), \tag{1.5}$$

which clearly now is a particular case of a general estimate for the Rayleigh quotient (cf. [19]),

$$\left|\frac{(x,Ax)}{(x,x)} - \frac{(y,Ay)}{(y,y)}\right| \leq (\lambda_{\max} - \lambda_{\min})\sin(\theta(x,y)), \tag{1.6}$$

where $A$ is a Hermitian operator and $\lambda_{\max} - \lambda_{\min}$ is the spread of its spectrum.

We show that the multidimensional analogue of (1.5) can be interpreted as a bound on the change in the Ritz values with the change of the trial subspace, in the particular case where the Rayleigh–Ritz method is applied to an orthogonal projector. We then extend it to prove a general result for Ritz values for an arbitrary Hermitian operator $A$, not necessarily a projector: let $\Lambda\left(P_{\mathcal{X}}A|_{\mathcal{X}}\right)$ be the vector of Ritz values in nonincreasing order for the operator $A$ on a trial subspace $\mathcal{X}$, which is perturbed to give another trial subspace $\mathcal{Y}$; then $|\Lambda\left(P_{\mathcal{X}}A|_{\mathcal{X}}\right)-\Lambda\left(P_{\mathcal{Y}}A|_{\mathcal{Y}}\right)| \prec_w (\lambda_{\max}-\lambda_{\min})\, \sin\Theta(\mathcal{X},\mathcal{Y})$, which is a multidimensional analogue of (1.6). Our present proof is based on a classical but rarely used idea of extending a Hermitian operator in $\mathcal{H}$ to an orthogonal projector in the "double" space $\mathcal{H}^2$, preserving its Ritz values.

An application of our Ritz values weak majorization result for Laplacian graph spectra comparison is suggested in section 5, based on the possibility of interpreting eigenvalues of the edge Laplacian of a given graph as Ritz values of the edge Laplacian of the complete graph. We prove that $\sum_k |\lambda_k^1-\lambda_k^2| \leq nl$, where $\lambda_k^1$ and $\lambda_k^2$ are all ordered elements of the Laplacian spectra of two graphs with the same $n$ vertices and with $l$ equal to the number of differing edges.

The rest of the paper is organized as follows. In section 2, we provide some background, definitions, and several statements concerning weak majorization, principal angles between subspaces, and extensions of Hermitian operators to projectors. In section 3, we prove in Theorems 3.2 and 3.3 that the absolute value of the change in (the squares of) the sines and cosines is weakly majorized by the sines of the angles between the original and perturbed subspaces. In section 4, we prove in Theorem 4.2 that a change in the Ritz values in the Rayleigh–Ritz method with respect to the change in the trial subspaces is weakly majorized by the sines of the principal angles between the original and perturbed trial subspaces times a constant. In section 5, we apply our Ritz values weak majorization result to Laplacian graphs spectra comparison. Section 6 gives brief conclusions.

This paper is related to several different subjects: majorization, principal angles, the Rayleigh–Ritz method, and Laplacian graph spectra. In most cases, whenever possible, we cite books rather than the original works in order to keep our already quite long list of references within a reasonable size.

## 2. Definitions and preliminaries.

In this section we introduce some definitions, basic concepts, and generally familiar results for later use.

### 2.1. Weak majorization.

Majorization is a well-known (e.g., see [14, pp. 45–49] or [24, pp. 9–14]) important mathematical concept with numerous applications.

For a real vector $x=[x_1,\ldots,x_n]$ let $x^{\downarrow}$ be the vector obtained by rearranging the entries of $x$ in an algebraically nonincreasing order, $x_1^{\downarrow}\geq\cdots\geq x_n^{\downarrow}$. We denote $[|x_1|,\ldots,|x_n|]$ by $|x|$. We say that vector $y$ weakly majorizes vector $x$, and we use the notation $[x_1,\ldots,x_n]\prec_w[y_1,\ldots,y_n]$ or $x\prec_w y$ if $\sum_{i=1}^k x_i^{\downarrow}\leq\sum_{i=1}^k y_i^{\downarrow}$ for $k=1,\ldots,n$. If in addition the sums above for $k=n$ are equal, $y$ (strongly) majorizes vector $x$, but we do not use this type of majorization in the present paper. Two vectors of different lengths may be compared by simply appending zeroes to increase the size of the smaller vector to make the vectors the same length.

Weak majorization is a powerful tool for bounds involving eigenvalues and singular values and is covered, e.g., in [10], [24], [1], and [15], which we follow here and to which we refer the reader for references to the original works and all necessary proofs. In the present paper, we use several well-known statements that we formulate for operators $\mathcal{H}\to\mathcal{H}$ and overview briefly below.

Let $S(A)$ denote the vector of all singular values of $A : \mathcal{H} \to \mathcal{H}$ in nonincreasing order, i.e., $S(A) = S^{\downarrow}(A)$, while individual singular values of $A$ enumerated in nonincreasing order are denoted by $s_i(A)$. For Hermitian $A$ let $\Lambda(A)$ denote the vector of all eigenvalues of $A$ in nonincreasing order, i.e., $\Lambda(A) = \Lambda^{\downarrow}(A)$, while individual eigenvalues of $A$ enumerated in nonincreasing order are denoted by $\lambda_i(A)$.

The starting point for weak majorization results that we use in this paper is, e.g., [24, Theorem 9.G.1, p. 241], given next.

THEOREM 2.1. $\Lambda(A+B) \prec_w \Lambda(A) + \Lambda(B)$ [illegible] $A$ [illegible] $B$

This follows easily from Ky Fan's trace maximum principle [24, Theorem 20.A.2, p. 511] and the fact that the maximum of a sum is bounded from above by the sum of the maxima. For general $A : \mathcal{H} \to \mathcal{H}$ and $B : \mathcal{H} \to \mathcal{H}$, it follows from Theorem 2.1, since the top half of the spectrum of the Hermitian 2-by-2 block operator $\left[\begin{smallmatrix} 0 & A \\ A^* & 0 \end{smallmatrix}\right]$ is nothing but $S(A)$, that (see, e.g., [15, Cor. 3.4.3, p. 196]) we have the following.

COROLLARY 2.2. $S(A \pm B) \prec_w S(A) + S(B)$

A more delicate and stronger result is the following Lidskii theorem (see, e.g., [1, Theorem III.4.1, p. 69]), which can be proved using the Wielandt maximum principle (e.g., [1, Theorem III.3.5, p. 67]).

THEOREM 2.3. [illegible] $A$ [illegible] $B$ [illegible] $1 \le i_1 < \cdots < i_k \le n = \dim\mathcal{H}$ [illegible] $\sum_{j=1}^{k} \lambda_{i_j}(A+B) \le \sum_{j=1}^{k} \lambda_{i_j}(A) + \sum_{j=1}^{k} \lambda_j(B)$, $k = 1, \ldots, n$.

By choosing an appropriate set of indices, Theorem 2.3 for Hermitian $A$ and $B$ immediately gives $\Lambda(A) - \Lambda(B) \prec_w \Lambda(A-B)$, which for singular values of arbitrary $A : \mathcal{H} \to \mathcal{H}$ and $B : \mathcal{H} \to \mathcal{H}$ is equivalent (see [1, section IV.3, pp. 98–101]) to, e.g., [15, Theorem 3.4.5, p. 198] or [1, Theorem IV.3.4, p. 100], as follows.

COROLLARY 2.4. $|S(A) - S(B)| \prec_w S(A-B)$

Applying Corollary 2.4 to properly shifted Hermitian operators, we get the next result.

COROLLARY 2.5. $|\Lambda(A) - \Lambda(B)| \prec_w S(A-B)$ [illegible] $A$ [illegible] $B$

We finally need the so-called pinching inequality (see, e.g., [10, Theorem II.5.1, p. 52] or [1, Problem II.5.5, p. 50]), which we write as follows.

THEOREM 2.6. [illegible] $P$ [illegible] $S(PAP \pm (I-P)A(I-P)) \prec_w S(A)$.

[illegible]. Indeed, $A = PAP + (I-P)A(I-P) + PA(I-P) + (I-P)AP$, so let $B = PAP + (I-P)A(I-P) - PA(I-P) - (I-P)AP$; then $(2P-I)A(2P-I) = B$, where $2P - I$ is unitary Hermitian, and thus $A^*A$ and $B^*B$ are similar and $S(A) = S(B)$. Evidently, $PAP + (I-P)A(I-P) = (A+B)/2$, and so the pinching result with the plus follows from Corollary 2.2. The pinching result with the minus is equivalent to the pinching result with the plus, since the sign does not change the singular values on the left-hand side: $S(PAP \pm (I-P)A(I-P)) = S(PAP) \cup S((I-P)A(I-P))$, since the ranges of $PAP$ and $(I-P)A(I-P)$ are disjoint. $\square$

**2.2. Principal angles between subspaces.** Let $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ be orthogonal projectors onto the subspaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, of the space $\mathcal{H}$. We define the set of cosines of principal angles between subspaces $\mathcal{X}$ and $\mathcal{Y}$ by

$$\cos\Theta(\mathcal{X},\mathcal{Y}) = [s_1(P_{\mathcal{X}}P_{\mathcal{Y}}), \ldots, s_m(P_{\mathcal{X}}P_{\mathcal{Y}})], \qquad m = \min\{\dim\mathcal{X}; \dim\mathcal{Y}\}. \tag{2.1}$$

Our definition (2.1) is evidently symmetric: $\Theta(\mathcal{X},\mathcal{Y}) = \Theta(\mathcal{Y},\mathcal{X})$. By definition, the cosines are arranged in nonincreasing order, i.e., $\cos(\Theta(\mathcal{X},\mathcal{Y})) = (\cos(\Theta(\mathcal{X},\mathcal{Y})))^{\downarrow}$, while the angles $\theta_i(\mathcal{X},\mathcal{Y}) \in [0, \pi/2]$, $i = 1, \ldots, m$, and their sines are in nondecreasing order.

The concept of principal angles is closely connected to cosine-sine (CS) decompositions of unitary operators; we refer the reader to the books [37], [38], [1] for the history and references to the original publications on principal angles and the CS decomposition. We need several simple but important statements about the angles, provided below. In the particular case $\dim\mathcal{X} = \dim\mathcal{Y}$, the standard CS decomposition can be used, and the statements are easy to derive. For the general case $\dim\mathcal{X} \neq \dim\mathcal{Y}$ that is necessary for us here, they can be obtained using the general (rectangular) form of the CS decomposition described, e.g., in [31], [30]. For completeness we provide the proofs here using ideas from [13], [7], preparing our work to be more easily extended to infinite dimensional Hilbert spaces.

In [18, Theorem 3.4, p. 2017 and Theorem 3.5, p. 2018], without proofs we formulate statements equivalent to the theorem below; see also [3, Example 1.2.6].

THEOREM 2.7. [illegible] $\pi/2$ [illegible]

$$\left[\frac{\pi}{2},\dots,\frac{\pi}{2},(\Theta(\mathcal{X},\mathcal{Y}))^{\downarrow}\right] = \left[\frac{\pi}{2}-\Theta(\mathcal{X},\mathcal{Y}^{\perp}),0,\dots,0\right], \tag{2.2}$$

[illegible] $\max\{\dim\mathcal{X}-\dim\mathcal{Y};0\}$ [illegible] $\pi/2$ [illegible]

[illegible]

$$[(\Theta(\mathcal{X},\mathcal{Y}))^{\downarrow},0,\dots,0] = [(\Theta(\mathcal{X}^{\perp},\mathcal{Y}^{\perp}))^{\downarrow},0,\dots,0], \tag{2.3}$$

[illegible] Let $\mathfrak{M}_{00} = \mathcal{X}\cap\mathcal{Y}$, $\mathfrak{M}_{01} = \mathcal{X}\cap\mathcal{Y}^{\perp}$, $\mathfrak{M}_{10} = \mathcal{X}^{\perp}\cap\mathcal{Y}$, $\mathfrak{M}_{11} = \mathcal{X}^{\perp}\cap\mathcal{Y}^{\perp}$, as suggested in [13, p. 381].

Each of the subspaces is invariant with respect to orthoprojectors $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ and their products, and so each of the subspaces contributes independently to the set of singular values of $P_{\mathcal{X}}P_{\mathcal{Y}}$ in (2.1). Specifically, there are $\dim\mathfrak{M}_{00}$ ones, $\dim\mathfrak{M}_0$ singular values in the interval $(0,1)$ equal to $\cos\Theta(\mathfrak{M}_0,\mathcal{Y})$, where $\mathfrak{M}_0 = \mathcal{X}\cap(\mathfrak{M}_{00}\oplus\mathfrak{M}_{01})^{\perp}$, and all other singular values are zeroes; thus,

$$(\Theta(\mathcal{X},\mathcal{Y}))^{\downarrow} = \left[\frac{\pi}{2},\dots,\frac{\pi}{2},(\Theta(\mathfrak{M}_0,\mathcal{Y}))^{\downarrow},0,\dots,0\right], \tag{2.4}$$

where there are $\min\{\dim(\mathfrak{M}_{01});\dim(\mathfrak{M}_{10})\}$ values $\pi/2$ and $\dim(\mathfrak{M}_{00})$ zeroes.

The subspace $\mathfrak{M}_0$ does not change if we substitute $\mathcal{Y}^{\perp}$ for $\mathcal{Y}$ in (2.4), so we have

$$(\Theta(\mathcal{X},\mathcal{Y}^{\perp}))^{\downarrow} = \left[\frac{\pi}{2},\dots,\frac{\pi}{2},(\Theta(\mathfrak{M}_0,\mathcal{Y}^{\perp}))^{\downarrow},0,\dots,0\right],$$

where there are $\min\{\dim(\mathfrak{M}_{00});\dim(\mathfrak{M}_{11})\}$ values $\pi/2$ and $\dim(\mathfrak{M}_{01})$ zeroes. Since $\lambda$ is an eigenvalue of $(P_{\mathcal{X}}P_{\mathcal{Y}})|_{\mathfrak{M}_0}$ if and only if $1-\lambda$ is an eigenvalue of $(P_{\mathcal{X}}P_{\mathcal{Y}^{\perp}})|_{\mathfrak{M}_0}$, we have $\frac{\pi}{2}-\Theta(\mathfrak{M}_0,\mathcal{Y}^{\perp}) = (\Theta(\mathfrak{M}_0,\mathcal{Y}))^{\downarrow}$, and the latter equality turns into

$$\frac{\pi}{2}-\Theta(\mathcal{X},\mathcal{Y}^{\perp}) = \left[\frac{\pi}{2},\dots,\frac{\pi}{2},(\Theta(\mathfrak{M}_0,\mathcal{Y}))^{\downarrow},0,\dots,0\right], \tag{2.5}$$

where there are $\dim(\mathfrak{M}_{01})$ values $\pi/2$, and $\min\{\dim(\mathfrak{M}_{00});\dim(\mathfrak{M}_{11})\}$ zeroes. To obtain (2.2), we make (2.4) and (2.5) equal by adding $\max\{\dim\mathfrak{M}_{01}-\dim\mathfrak{M}_{10};0\}$ values $\pi/2$ to (2.4) and $\max\{\dim\mathfrak{M}_{00}-\dim\mathfrak{M}_{11};0\}$ zeroes to (2.5), and by noting that, since $\dim(\mathcal{X}\cap\mathcal{Y}^\perp) = \dim\mathcal{X} + \dim\mathcal{Y}^\perp - \dim(\mathcal{X}+\mathcal{Y}^\perp)$, $\mathcal{X}^\perp\cap\mathcal{Y} = (\mathcal{X}+\mathcal{Y}^\perp)^\perp$ and $\dim\mathcal{Y}^\perp - \dim((\mathcal{X}+\mathcal{Y}^\perp)^\perp) = \dim(\mathcal{X}+\mathcal{Y}^\perp) - \dim\mathcal{Y}$, we have $\dim\mathfrak{M}_{01} - \dim\mathfrak{M}_{10} = \dim(\mathcal{X}\cap\mathcal{Y}^\perp) - \dim(\mathcal{X}^\perp\cap\mathcal{Y}) = \dim\mathcal{X} + \dim\mathcal{Y}^\perp - \dim(\mathcal{X}+\mathcal{Y}^\perp) - \dim((\mathcal{X}+\mathcal{Y}^\perp)^\perp) = \dim\mathcal{X} - \dim\mathcal{Y}$.

The proof above shows that there are $\dim\mathfrak{M}_{00} = \dim(\mathcal{X}\cap\mathcal{Y})$ zeroes on the right-hand side in (2.2). To prove (2.3), we substitute $\mathcal{X}^\perp$ for $\mathcal{X}$ in (2.2) to get $\left[\frac{\pi}{2},\ldots,\frac{\pi}{2},(\Theta(\mathcal{X}^\perp,\mathcal{Y}))^\downarrow\right] = \left[\frac{\pi}{2} - \Theta(\mathcal{X}^\perp,\mathcal{Y}^\perp),0,\ldots,0\right]$ with $\dim(\mathcal{X}^\perp\cap\mathcal{Y})$ zeroes on the right, on the one hand, and exchange $\Theta(\mathcal{X},\mathcal{Y}) = \Theta(\mathcal{Y},\mathcal{X})$ in (2.2) and then substitute $\mathcal{X}^\perp$ for $\mathcal{X}$ to obtain $\left[\frac{\pi}{2},\ldots,\frac{\pi}{2},(\Theta(\mathcal{Y},\mathcal{X}^\perp))^\downarrow\right] = \left[\frac{\pi}{2} - \Theta(\mathcal{Y},\mathcal{X}),0,\ldots,0\right]$ with $\dim(\mathcal{Y}\cap\mathcal{X}^\perp)$ zeroes on the right, on the other hand. We have equal numbers of zeroes on the right in both equalities, and $\Theta(\mathcal{X}^\perp,\mathcal{Y}) = \Theta(\mathcal{Y},\mathcal{X}^\perp)$ by the symmetry of our definition (2.1), so subtracting both equalities from $\pi/2$ leads to (2.3). □

We also use the following trivial, but crucial, statement.

LEMMA 2.8. $\Lambda\left((P_\mathcal{X}P_\mathcal{Y})|_\mathcal{X}\right) = [\cos^2\Theta(\mathcal{X},\mathcal{Y}),0,\ldots,0]$, [illegible] $\max\{\dim\mathcal{X}-\dim\mathcal{Y};0\}$ [illegible]

[illegible] The operator $(P_\mathcal{X}P_\mathcal{Y})|_\mathcal{X} = \left((P_\mathcal{X}P_\mathcal{Y})(P_\mathcal{X}P_\mathcal{Y})^\star\right)|_\mathcal{X}$ is Hermitian nonnegative definite, and its spectrum can be represented using the definition of angles (2.1). The number of extra zeroes is exactly the difference between the number $\dim\mathcal{X}$ of Ritz values and the number $\min\{\dim\mathcal{X};\dim\mathcal{Y}\}$ of principal angles. □

Finally, we need the following characterization of singular values of the difference of projectors, which for $\dim\mathcal{X} = \dim\mathcal{Y}$ appears, e.g., in [38, Theorem 5.5.5, p. 43].

THEOREM 2.9.

$$[S(P_\mathcal{X} - P_\mathcal{Y}),0,\ldots,0] = [1,\ldots,1,(\sin\Theta(\mathcal{X},\mathcal{Y}),\sin\Theta(\mathcal{X},\mathcal{Y}))^\downarrow,0,\ldots,0],$$

[illegible] $|\dim\mathcal{X}-\dim\mathcal{Y}|$ [illegible] 1 [illegible] $\sin\Theta(\mathcal{X},\mathcal{Y})$ [illegible] 0 [illegible]

[illegible] The projectors $P_\mathcal{X}$ and $P_\mathcal{Y}$ are idempotent, which implies, on the one hand,

$$(P_\mathcal{X} - P_\mathcal{Y})^2 = P_\mathcal{X}(I - P_\mathcal{Y}) + P_\mathcal{Y}(I - P_\mathcal{X}) = P_\mathcal{X}P_{\mathcal{Y}^\perp} + P_\mathcal{Y}P_{\mathcal{X}^\perp},$$

so that the subspace $\mathcal{X}$ is invariant under $(P_\mathcal{X} - P_\mathcal{Y})^2$. On the other hand,

$$(P_\mathcal{X} - P_\mathcal{Y})^2 = (I - P_\mathcal{X})P_\mathcal{Y} + (I - P_\mathcal{Y})P_\mathcal{X} = P_{\mathcal{X}^\perp}P_\mathcal{Y} + P_{\mathcal{Y}^\perp}P_\mathcal{X},$$

so that the subspace $\mathcal{X}^\perp$ is also invariant under $(P_\mathcal{X} - P_\mathcal{Y})^2$. The projectors $P_\mathcal{X}$ and $P_\mathcal{Y}$ are orthogonal, and thus the operator $(P_\mathcal{X} - P_\mathcal{Y})^2$ is Hermitian, and its spectrum can be represented as a union (counting the multiplicities) of the spectra of its restrictions to the complementary invariant subspaces $\mathcal{X}$ and $\mathcal{X}^\perp$:

$$\Lambda\left((P_\mathcal{X} - P_\mathcal{Y})^2\right) = [\Lambda((P_\mathcal{X}P_{\mathcal{Y}^\perp})|_\mathcal{X}),\Lambda((P_{\mathcal{X}^\perp}P_\mathcal{Y}))|_{\mathcal{X}^\perp}]^\downarrow.$$

Using Lemma 2.8 and statement (2.2) of Theorem 2.7,

$$\begin{aligned}[\Lambda((P_\mathcal{X}P_{\mathcal{Y}^\perp})|_\mathcal{X}),0,\ldots,0] &= [\cos^2\Theta(\mathcal{X},\mathcal{Y}^\perp),0,\ldots,0]\\ &= [1,\ldots,1,(\sin^2\Theta(\mathcal{X},\mathcal{Y}))^\downarrow,0,\ldots,0],\end{aligned}$$

where there are $\max\{\dim\mathcal{X}-\dim\mathcal{Y};0\}$ leading 1's and possibly extra 0's to match the sizes, and

$$\begin{aligned}[\Lambda((P_{\mathcal{X}^\perp}P_\mathcal{Y})|_{\mathcal{X}^\perp}),0,\ldots,0] &= [\cos^2\Theta(\mathcal{X}^\perp,\mathcal{Y}),0,\ldots,0]\\ &= [1,\ldots,1,(\sin^2\Theta(\mathcal{X},\mathcal{Y}))^\downarrow,0,\ldots,0],\end{aligned}$$

where there are $\max\{\dim\mathcal{Y}-\dim\mathcal{X};0\}$ leading 1's and possibly extra 0's to match the sizes. Combining these two relations and taking the square root completes the proof. $\square$

**2.3. Extending operators to isometries and projectors.** In this subsection we present a simple and known technique (see, e.g., [12] and [35, p. 461]) for extending a Hermitian operator to a projector. We give an alternative proof based on extending an arbitrary normalized operator $B$ to an isometry $\hat{B}$ (in matrix terms, a matrix with orthonormal columns). [9, Problem X.1.26, p. 455] and [1, Problem I.3.6, p. 11] extend $B$ to a block 2-by-2 unitary operator. Our technique is similar and results in a 2-by-1 isometry operator $\hat{B}$ that coincides with the first column of the 2-by-2 unitary extension.

LEMMA 2.10. [illegible] $B:\mathcal{H}\to\mathcal{H}$ [illegible] 2 [illegible] 1 [illegible] $\hat{B}:\mathcal{H}\to\mathcal{H}^2$ [illegible] $\hat{B}$ [illegible] $B$

[illegible] $B^*B$ is Hermitian nonnegative definite, and all its eigenvalues are bounded by one, since all singular values of $B$ are bounded by one. Therefore, $I-B^*B$ is Hermitian and nonnegative definite, and thus possesses a Hermitian nonnegative definite square root. Let

$$\hat{B}=\left[\begin{array}{c} B \\ \sqrt{I-B^*B} \end{array}\right].$$

By direct calculation, $\hat{B}^*\hat{B}=B^*B+\sqrt{I-B^*B}\sqrt{I-B^*B}=I$; i.e., $\hat{B}$ is an isometry. $\square$

Now we use Lemma 2.10 to extend, in a similar sense, a shifted and normalized Hermitian operator to an orthogonal projector.

THEOREM 2.11 (see [12] and [35, p. 461]). [illegible] $A:\mathcal{H}\to\mathcal{H}$ [illegible] $[0,1]$ [illegible] 2 [illegible] 2 [illegible] $\hat{A}:\mathcal{H}^2\to\mathcal{H}^2$ [illegible] $A$

[illegible] There exists $\sqrt{A}$, which is also Hermitian and has its eigenvalues enclosed in $[0,1]$. Applying Lemma 2.10 to $B=\sqrt{A}$, we construct the isometry $\hat{B}$ and set

$$\hat{A}=\hat{B}\hat{B}^*=\left[\begin{array}{c} \sqrt{A} \\ \sqrt{I-A} \end{array}\right]\left[\begin{array}{cc} \sqrt{A} & \sqrt{I-A} \end{array}\right]=\left[\begin{array}{cc} A & \sqrt{A(I-A)} \\ \sqrt{A(I-A)} & I-A \end{array}\right].$$

We see that indeed the upper left block is equal to $A$. We can use the fact that $\hat{B}$ is an isometry to show that $\hat{A}$ is an orthogonal projector, or that can be checked directly by calculating $\hat{A}^2=\hat{A}$ and noticing that $\hat{A}$ is Hermitian by construction. $\square$

Introducing $S=\sqrt{A}$ and $C=\sqrt{I-A}$, we obtain

$$\hat{A}=\left[\begin{array}{cc} S^2 & SC \\ SC & C^2 \end{array}\right],$$

which is a well-known (see, e.g., [13], [6]) block form of an orthogonal projector that can alternatively be derived using the CS decomposition of unitary operators; see, e.g., [37], [38], [1].

The importance of Theorem 2.11 can be better seen if we reformulate it as follows.

THEOREM 2.12 (cf. [9, Example X.27, p. s455]). [illegible] $A:\mathcal{H}\to\mathcal{H}$ [illegible] $[0,1]$ [illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible] $\mathcal{H}^2$ [illegible] $A$ [illegible] $(P_{\mathcal{X}}P_{\mathcal{Y}})|_{\mathcal{X}}$ [illegible] $P_{\mathcal{X}}$ [illegible] $P_{\mathcal{Y}}$ [illegible] $\mathcal{H}^2$ [illegible] $|_{\mathcal{X}}$ [illegible] $\mathcal{X}$

[illegible]. We use Theorem 2.11 and take $P_{\mathcal{Y}} = \hat{A}$ and $P_{\mathcal{X}} = \left[\begin{smallmatrix} I & 0 \\ 0 & 0 \end{smallmatrix}\right]$. ☐

Similar to Theorem 2.12, Lemma 2.10 implicitly states that an arbitrary normalized operator $B$ is unitarily equivalent to a product of the partial isometry $\hat{B}$ in $\mathcal{H}^2$ and the orthogonal projector in $\mathcal{H}^2$ that selects the upper block in $\hat{B}$ (called $P_{\mathcal{X}}$ in the proof of Theorem 2.12). It is instructive to compare this product to the classical polar decomposition of $B$ that is a product of a partial isometry and a Hermitian nonnegative definite operator in $\mathcal{H}$. In $\mathcal{H}^2$, we can choose the second factor to be an orthogonal projector! This statement together with Theorem 2.12 can provide interesting canonical decompositions in $\mathcal{H}^2$ that apparently are not exploited at present, but in our opinion deserve attention.

We take advantage of Theorem 2.12 in the present paper. Using Lemma 2.8 with (2.1), Theorem 2.12 implies that [illegible]. This surprising idea appears to be very powerful. It allows us, in section 4, to obtain a novel result on the sensitivity of Ritz values with respect to the trial subspace by reducing the investigation of the Rayleigh–Ritz method to the analysis of the principal angles between subspaces that we provide in the next section.

## 3. Majorization for angles.

In this section we prove the main results of the present paper involving sines and cosines and their squares of principal angles, but we start with a known statement that involves the principal angles themselves.

THEOREM 3.1 (see [34, Theorem 3.2, p. 514]). [illegible] $\mathcal{X}$ $\mathcal{Y}$ [illegible] $\mathcal{Z}$ [illegible]

$$|\Theta(\mathcal{X}, \mathcal{Z}) - \Theta(\mathcal{Y}, \mathcal{Z})| \prec_w \Theta(\mathcal{X}, \mathcal{Y}). \tag{3.1}$$

Theorem 3.1 deals with the principal angles themselves, and the obvious question is: Are there similar results for a function of these angles, in particular for sines and cosines and their squares? For one dimensional subspaces, estimate (3.1) turns into (1.1), which, as discussed in the Introduction, implies the estimate (1.2) for the sine. According to an anonymous referee, it appears to be known to some specialists that the same inference can be made for tuples of angles, but there is no good reference for this at present. Below we give easy direct proofs in a unified way for the sines and cosines and their squares.

We first prove the estimates for sine and cosine, which are straightforward generalizations of the 1D sine (1.2) and cosine (1.3) inequalities from the Introduction.

THEOREM 3.2. [illegible] $\dim\mathcal{X} = \dim\mathcal{Y}$. [illegible]

$$|\sin\Theta(\mathcal{X}, \mathcal{Z}) - \sin\Theta(\mathcal{Y}, \mathcal{Z})| \prec_w \sin\Theta(\mathcal{X}, \mathcal{Y}), \tag{3.2}$$

$$|\cos\Theta(\mathcal{X}, \mathcal{Z}) - \cos\Theta(\mathcal{Y}, \mathcal{Z})| \prec_w \sin\Theta(\mathcal{X}, \mathcal{Y}). \tag{3.3}$$

[illegible]. Let $P_{\mathcal{X}}$, $P_{\mathcal{Y}}$, and $P_{\mathcal{Z}}$ be the corresponding orthogonal projectors onto the subspaces $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, respectively. We prove the sine estimate (3.2), using the idea of [33]. Starting with $(P_{\mathcal{X}} - P_{\mathcal{Z}}) - (P_{\mathcal{Y}} - P_{\mathcal{Z}}) = P_{\mathcal{X}} - P_{\mathcal{Y}}$, as in the proof of the 1D sine estimate (1.2), we use Corollary 2.4 to obtain

$$|S(P_{\mathcal{X}} - P_{\mathcal{Z}}) - S(P_{\mathcal{Y}} - P_{\mathcal{Z}})| \prec_w S(P_{\mathcal{X}} - P_{\mathcal{Y}}).$$

The singular values of the difference of two orthoprojectors are described by Theorem 2.9. Since $\dim\mathcal{X} = \dim\mathcal{Y}$, we have the same number of extra 1's up front in $S(P_{\mathcal{X}} - P_{\mathcal{Z}})$

and in $S(P_{\mathcal{Y}}-P_{\mathcal{Z}})$, so that the extra 1's are canceled and the set of nonzero entries of $|S(P_{\mathcal{X}}-P_{\mathcal{Z}})-S(P_{\mathcal{Y}}-P_{\mathcal{Z}})|$ consists of nonzero entries of $|\sin\Theta(\mathcal{X},\mathcal{Z})-\sin\Theta(\mathcal{Y},\mathcal{Z})|$ repeated twice. The nonzero entries of $S(P_{\mathcal{X}}-P_{\mathcal{Y}})$ are by Theorem 2.9 the nonzero entries of $\sin\Theta(\mathcal{X},\mathcal{Y})$ also repeated twice, and thus we come to (3.2).

The cosine estimate (3.3) follows directly from the sine estimate (3.2) with $\mathcal{Z}^{\perp}$ instead of $\mathcal{Z}$ because of (2.2), and utilizing the assumption $\dim\mathcal{X}=\dim\mathcal{Y}$. □

In our earlier paper [18, Lemma 5.1, p. 2023 and Lemma 5.2, p. 2025] we obtained a particular case of Theorem 3.2, only for the largest change in the sine and the cosine, but with improved constants. We are not presently able, however, to modify the proofs of [18] using weak majorization, in order to improve the estimates of Theorem 3.2 by introducing these same constants.

Our last, but not least, result in this series is the weak majorization bound for the sines or cosines squared, which provides the foundation for the rest of the paper.

THEOREM 3.3. $\dim\mathcal{X}=\dim\mathcal{Y}$.

$$|\cos^2\Theta(\mathcal{X},\mathcal{Z})-\cos^2\Theta(\mathcal{Y},\mathcal{Z})|=|\sin^2\Theta(\mathcal{X},\mathcal{Z})-\sin^2\Theta(\mathcal{Y},\mathcal{Z})|\prec_w\sin\Theta(\mathcal{X},\mathcal{Y}).$$

The equality is evident. To prove the majorization result for the sines squared, we start with the useful pinching identity

$$(P_{\mathcal{X}}-P_{\mathcal{Z}})^2-(P_{\mathcal{Y}}-P_{\mathcal{Z}})^2=P_{\mathcal{Z}^{\perp}}(P_{\mathcal{X}}-P_{\mathcal{Y}})P_{\mathcal{Z}^{\perp}}-P_{\mathcal{Z}}(P_{\mathcal{X}}-P_{\mathcal{Y}})P_{\mathcal{Z}}.$$

Applying Corollary 2.4, we obtain

$$\left|S\left((P_{\mathcal{X}}-P_{\mathcal{Z}})^2\right)-S\left((P_{\mathcal{Y}}-P_{\mathcal{Z}})^2\right)\right|\prec_w S\left(P_{\mathcal{Z}^{\perp}}(P_{\mathcal{X}}-P_{\mathcal{Y}})P_{\mathcal{Z}^{\perp}}-P_{\mathcal{Z}}(P_{\mathcal{X}}-P_{\mathcal{Y}})P_{\mathcal{Z}}\right).$$

For the left-hand side we use Theorem 2.9, as in the proof of Theorem 3.2, except that we are now working with the squares. For the right-hand side, the pinching Theorem 2.6 gives

$$S\left(P_{\mathcal{Z}^{\perp}}(P_{\mathcal{X}}-P_{\mathcal{Y}})P_{\mathcal{Z}^{\perp}}-P_{\mathcal{Z}}(P_{\mathcal{X}}-P_{\mathcal{Y}})P_{\mathcal{Z}}\right)\prec_w S\left(P_{\mathcal{X}}-P_{\mathcal{Y}}\right),$$

and we use Theorem 2.9 again to characterize $S\left(P_{\mathcal{X}}-P_{\mathcal{Y}}\right)$, as in the proof of Theorem 3.2. □

**4. Changes in the trial subspace in the Rayleigh–Ritz method.** In this section, we explore a simple, but deep, connection between the principal angles and the Rayleigh–Ritz method that we discuss in the Introduction. We demonstrate that the analysis of the influence of changes in a trial subspace in the Rayleigh–Ritz method is a natural extension of the theory concerning principal angles and the proximity of two subspaces developed in the previous section.

For the reader's convenience, let us repeat here the definition of Ritz values from the Introduction: Let $A:\mathcal{H}\to\mathcal{H}$ be a Hermitian operator and $P_{\mathcal{X}}$ be an orthogonal projector to a subspace $\mathcal{X}$ of $\mathcal{H}$. The eigenvalues $\Lambda(P_{\mathcal{X}}A|_{\mathcal{X}})$ are the Ritz values of operator $A$ with respect to $\mathcal{X}$, which is called the trial subspace.

Let $\mathcal{X}$ and $\mathcal{Y}$ both be subspaces of $\mathcal{H}$ and $\dim\mathcal{X}=\dim\mathcal{Y}$. The goal of this section is to analyze the sensitivity of Ritz values with respect to the trial subspaces, specifically to bound the change $|\Lambda\left(P_{\mathcal{X}}A|_{\mathcal{X}}\right)-\Lambda\left(P_{\mathcal{Y}}A|_{\mathcal{Y}}\right)|$ in terms of $\sin\Theta(\mathcal{X},\mathcal{Y})$ using weak majorization. Such an estimate is already obtained in Theorem 10 of our earlier paper [19] by applying Corollary 2.5 to the matrices of $P_{\mathcal{X}}A|_{\mathcal{X}}$ and $P_{\mathcal{Y}}A|_{\mathcal{Y}}$. This approach, however, leads to an extra factor $\sqrt{2}$ on the right-hand side, which is conjectured in [19] to be artificial.

We remove this $\sqrt{2}$ factor in our new Theorem 4.2 by using the entirely different and novel approach: We connect the Ritz values with extension Theorems 2.11 and 2.12, on the one hand, and with the cosine squared of principal angles in Lemma 2.8, on the other hand. We have shown in Theorem 2.11 that a Hermitian nonnegative definite contraction operator can be extended to an orthogonal projector in a larger space. The extension has an extra nice property: It preserves the Ritz values.

Corollary 4.1. [illegible] 2.11 [illegible] $A : \mathcal{H} \to \mathcal{H}$ [illegible] $\mathcal{X} \subset H$ [illegible] $\hat{A} : \mathcal{H}^2 \to \mathcal{H}^2$ [illegible]

$$\hat{\mathcal{X}} = \begin{bmatrix} \mathcal{X} \\ 0 \end{bmatrix} \subset \hat{\mathcal{H}} = \begin{bmatrix} \mathcal{H} \\ 0 \end{bmatrix} \subset \mathcal{H}^2.$$

[illegible] Let $P_{\hat{\mathcal{H}}} : \mathcal{H}^2 \to \mathcal{H}^2$ be an orthogonal projector on the subspace $\hat{\mathcal{H}}$ and $P_{\hat{\mathcal{X}}} : \mathcal{H}^2 \to \mathcal{H}^2$ be an orthogonal projector on the subspace $\hat{\mathcal{X}}$. We use the equality sign to denote the trivial isomorphism between $\mathcal{H}$ and $\hat{\mathcal{H}}$; i.e., we simply write $\mathcal{H} = \hat{\mathcal{H}}$ and $\mathcal{X} = \hat{\mathcal{X}}$.

In this notation, we first observe that $A = P_{\hat{\mathcal{H}}} \hat{A}|_{\hat{\mathcal{H}}}$; i.e., the operator $A$ itself can be viewed as a result of the Rayleigh–Ritz method applied to the operator $\hat{A}$ in the trial subspace $\hat{\mathcal{H}}$. Second, we use the fact that a recursive application of the Rayleigh–Ritz method on a system of enclosed trial subspaces is equivalent to a direct single application of the Rayleigh–Ritz method to the smallest trial subspace; indeed, in our notation, $P_{\hat{\mathcal{H}}} P_{\hat{\mathcal{X}}} = P_{\hat{\mathcal{X}}} P_{\hat{\mathcal{H}}} = P_{\hat{\mathcal{X}}}$, since $\hat{\mathcal{X}} \subset \hat{\mathcal{H}}$, and thus

$$P_{\mathcal{X}} A|_{\mathcal{X}} = \left( P_{\hat{\mathcal{X}}} P_{\hat{\mathcal{H}}} \hat{A}|_{\hat{\mathcal{H}}} \right) \Big|_{\hat{\mathcal{X}}} = P_{\hat{\mathcal{X}}} \hat{A}|_{\hat{\mathcal{X}}}. \qquad \square$$

Next we note that Lemma 2.8 states that the Rayleigh–Ritz method applied to an orthogonal projector produces Ritz values, which are essentially the cosines squared of the principal angles between the range of the projector and the trial subspace. For the reader's convenience we reformulate Lemma 2.8 here: Let the Rayleigh–Ritz method be applied to $A = P_{\mathcal{Z}}$, where $P_{\mathcal{Z}}$ is an orthogonal projector onto a subspace $\mathcal{Z}$, and let $\mathcal{X}$ be the trial subspace in the Rayleigh–Ritz method; then the set of the Ritz values is $\Lambda\left(P_{\mathcal{X}} P_{\mathcal{Z}}|_{\mathcal{X}}\right) = [\cos^2 \Theta(\mathcal{X}, \mathcal{Z}), 0, \ldots, 0]$ with $\max\{\dim\mathcal{X} - \dim\mathcal{Z}; 0\}$ extra zeroes.

Now we are ready to direct our attention to the main topic of this section: the influence of changes in a trial subspace in the Rayleigh–Ritz method on the Ritz values.

Theorem 4.2. [illegible] $A : \mathcal{H} \to \mathcal{H}$ [illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible] $\mathcal{H}$ [illegible] $\dim\mathcal{X} = \dim\mathcal{Y}$ [illegible]

$$|\Lambda\left(P_{\mathcal{X}} A|_{\mathcal{X}}\right) - \Lambda\left(P_{\mathcal{Y}} A|_{\mathcal{Y}}\right)| \prec_w (\lambda_{\max} - \lambda_{\min}) \; \sin\Theta(\mathcal{X}, \mathcal{Y}), \tag{4.1}$$

[illegible] $\lambda_{\min}$ [illegible] $\lambda_{\max}$ [illegible] $A$ [illegible]

[illegible] We prove Theorem 4.2 in two steps. First we show that we can assume that $A$ is a nonnegative definite contraction without losing generality. Second, under these assumptions, we extend the operator $A$ to an orthogonal projector by Theorem 2.11 and use the facts that such an extension does not affect the Ritz values by Corollary 4.1 and that the Ritz values of an orthogonal projector can be interpreted as the cosines squared of principal angles between subspaces by Lemma 2.8, thus reducing the problem to the already established result on weak majorization of the cosine squared Theorem 3.3.

We observe that the statement of the theorem is invariant with respect to a shift and a scaling; indeed, for real $\alpha$ and $\beta$ if the operator $A$ is replaced with $\beta(A-\alpha)$ and $\lambda_{\min}$ and $\lambda_{\max}$ are correspondingly updated, both sides of (4.1) are just multiplied by $\beta$, and (4.1) is thus invariant with respect to $\alpha$ and $\beta$. Choosing $\alpha = \lambda_{\min}$ and $\beta = 1/(\lambda_{\max}-\lambda_{\min})$, the transformed operator $(A-\lambda_{\min})/(\lambda_{\max}-\lambda_{\min})$ is Hermitian with its eigenvalues enclosed in a segment $[0,1]$, and thus the statement (4.1) of the theorem can be equivalently rewritten as

$$|\Lambda\left(P_{\mathcal{X}}A|_{\mathcal{X}}\right) - \Lambda\left(P_{\mathcal{Y}}A|_{\mathcal{Y}}\right)| \prec_w \sin\Theta(\mathcal{X},\mathcal{Y}), \tag{4.2}$$

where we from now on assume that $A$ is a nonnegative definite contraction without losing generality.

The second step of the proof is to recast the problem into an equivalent problem for an orthogonal projector with the same Ritz values and principal angles. By Theorem 2.11 we can extend the nonnegative definite contraction $A$ to an orthogonal projector $P_{\hat{\mathcal{Z}}}$, where $\hat{\mathcal{Z}}$ is a subspace of $\mathcal{H}^2$. $P_{\hat{\mathcal{Z}}}$ has by Corollary 4.1 the same Ritz values with respect to trial subspaces,

$$\hat{\mathcal{X}} = \begin{bmatrix} \mathcal{X} \\ 0 \end{bmatrix} \subset \hat{\mathcal{H}} = \begin{bmatrix} \mathcal{H} \\ 0 \end{bmatrix} \subset \mathcal{H}^2 \quad \text{and} \quad \hat{\mathcal{Y}} = \begin{bmatrix} \mathcal{Y} \\ 0 \end{bmatrix} \subset \hat{\mathcal{H}} = \begin{bmatrix} \mathcal{H} \\ 0 \end{bmatrix} \subset \mathcal{H}^2,$$

as $A$ has with respect to the trial subspaces $\mathcal{X}$ and $\mathcal{Y}$. By Lemma 2.8, these Ritz values are equal to the cosines squared of the principal angles between $\hat{\mathcal{Z}}$ and the trial subspace $\hat{\mathcal{X}}$ or $\hat{\mathcal{Y}}$, possibly with the same number of zeroes being added. Moreover, the principal angles between $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ in $\mathcal{H}^2$ are clearly the same as those between $\mathcal{X}$ and $\mathcal{Y}$ in $\mathcal{H}$ and $\dim\hat{\mathcal{X}} = \dim\mathcal{X} = \dim\mathcal{Y} = \dim\hat{\mathcal{Y}}$. Thus, (4.2) can be equivalently reformulated as

$$|\cos^2\Theta(\hat{\mathcal{X}},\hat{\mathcal{Z}}) - \cos^2\Theta(\hat{\mathcal{Y}},\hat{\mathcal{Z}})| \prec_w \sin\Theta(\hat{\mathcal{X}},\hat{\mathcal{Y}}). \tag{4.3}$$

Finally, we notice that (4.3) is already proved in Theorem 3.3. □

REMARK 4.1. [illegible] [19, Remark 7], [illegible] $\lambda_{\max}-\lambda_{\min}$ [illegible] 4.2 [illegible]

$$\max_{x\in\mathcal{X}+\mathcal{Y},\, \|x\|=1}(x,Ax) - \min_{x\in\mathcal{X}+\mathcal{Y},\, \|x\|=1}(x,Ax),$$

[illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible]

REMARK 4.2. [illegible] 4.2 [illegible] $m = \dim\mathcal{X} = \dim\mathcal{Y}$ [illegible] $\alpha_1 \geq \cdots \geq \alpha_m$ [illegible] $A$ [illegible] $\mathcal{X}$ [illegible] $\beta_1 \geq \cdots \geq \beta_m$ [illegible] $A$ [illegible] $\mathcal{Y}$ [illegible] 4.2 [illegible]

$$\sum_{i=1}^{k} |\alpha_i - \beta_i|^{\downarrow} \leq (\lambda_{\max}-\lambda_{\min}) \sum_{i=1}^{k} \sin(\Theta_i(\mathcal{X},\mathcal{Y}))^{\downarrow}, \qquad k = 1,\ldots,m;$$

[illegible] $k = m$ [illegible]

$$\sum_{i=1}^{m} |\alpha_i - \beta_i| \leq (\lambda_{\max}-\lambda_{\min}) \sum_{i=1}^{m} \sin(\Theta_i(\mathcal{X},\mathcal{Y})), \tag{4.4}$$

[illegible] $k = 1$ [illegible]

$$\max_{j=1,\dots,m} |\alpha_j - \beta_j| \le (\lambda_{\max} - \lambda_{\min})\operatorname{gap}(\mathcal{X}, \mathcal{Y}), \tag{4.5}$$

[illegible] $\operatorname{gap}(\mathcal{X}, \mathcal{Y})$ [illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible] $\mathcal{X}$ [illegible] $\mathcal{Y}$ [illegible] (4.5) [illegible] [19, Theorem 5].

[illegible] $x$ [illegible] $y$ [illegible] $x \prec_w y$ [illegible] $\sum_{i=1}^n \phi(x_i) \le \sum_{i=1}^n \phi(y_i)$ [illegible] $\phi$ [illegible] [24, Proposition 4.B.2., p. 109]. [illegible] $\phi(t) = t^p$ [illegible] $p \ge 1$ [illegible] 4.2 [illegible]

$$\left(\sum_{i=1}^m |\alpha_i - \beta_i|^p\right)^{\frac{1}{p}} \le (\lambda_{\max} - \lambda_{\min}) \left(\sum_{i=1}^m \sin(\Theta_i(\mathcal{X}, \mathcal{Y}))^p\right)^{\frac{1}{p}}, \quad 1 \le p < \infty.$$

We finally note that the results of Theorem 4.2 are not intended for the case where one of the subspaces $\mathcal{X}$ or $\mathcal{Y}$ is invariant with respect to operator $A$. In such a case, it is natural to expect a much better bound that involves the square of the $\sin\Theta(\mathcal{X}, \mathcal{Y})$. Majorization results of this kind are not apparently known in the literature. Without majorization, estimates are available just for the largest change in the Ritz values; e.g., see [19], [21].

## 5. Application of the majorization results to graph spectra comparison.

In this section, we show that our majorization results can be applied to compare graph spectra. The graph spectra comparison can be used for graph matching and has applications in data mining; cf. [23].

The section is divided into three subsections. In subsection 5.1, we give all necessary definitions and basic facts concerning Laplacian graph spectra. In subsection 5.2, we connect the Laplacian graph spectrum and Ritz values, by introducing the graph edge Laplacian. Finally, in subsection 5.3, we prove our main result on the Laplacian graph spectra comparison.

### 5.1. Incidence matrices and graph Laplacians.

Here, we give mostly well-known relevant definitions (see, e.g., [5], [4], [26], [27], [28], [29]), just slightly tailored for our specific needs.

Let $V$ be a finite ordered set (of vertices), with an individual element (vertex) denoted by $v_i \in V$. Let $E_c$ be the finite ordered set (of all edges), with an individual element (edge) denoted by $e_k \in E_c$ such that every $e_k = [v_i, v_j]$ for all possible $i > j$. $E_c$ can be viewed as the set of edges of a complete simple graph with vertices $V$ (without self-loops and/or multiple edges). The results of the present paper are invariant with respect to specific ordering of vertices and edges.

Let $w_c : E_c \to \mathbf{R}$ be a function describing edge weights, i.e., $w_c(e_k) \in \mathbf{R}$. If for some edge $e_k$ the weight is positive, $w_c(e_k) > 0$, we call this edge present; if $w_c(e_k) = 0$, we say that the edge is absent. In this paper we do not allow negative edge weights. For a given weight function $w_c$, we define $E \subseteq E_c$ such that $e_k \in E$ if $w_c(e_k) \neq 0$, and we define $w$ to be the restriction of $w_c$ on all present edges $E$; i.e., $w$ is made of all nonzero values of $w_c$. A pair of sets of vertices $V$ and present edges $E$ with weights $w$ is called a graph $(V, E)$ or a weighted graph $(V, E, w)$.

The vertex-edge incidence matrix $Q_c$ of a complete graph $(V, E_c)$ is a matrix which has a row for each vertex and a column for each edge, with columnwise entries determined as $q_{ik} = 1$, $q_{jk} = -1$ for every edge $e_k = [v_i, v_j]$, $i > j$, in $E_c$ and with all other entries of $Q_c$ equal to zero. The vertex-edge incidence matrix $Q$ of a graph

$(V, E)$ is determined in the same way, but only for the edges present in $E$. The vertex-edge incidence matrix can be viewed as a matrix representation of a graph analogue of the divergence operator from partial differential equations (PDE).

Extending the PDE analogy, the matrix $L = QQ^*$ is called the graph Laplacian. In the PDE context, this definition corresponds to the negative Laplacian with the natural boundary conditions; cf. [25]. Let us note that in the graph theory literature such a definition of the graph Laplacian is usually attributed to directed graphs, even though reversing any edge direction does not affect the graph Laplacian.

If we want to take into account the weights, we can work with the matrix $Q \operatorname{diag}(w(E))Q^*$, which is an analogue of an isotropic diffusion operator, or we can introduce a more general edge matrix $W$ and work with $QWQ^*$, which corresponds to a general anisotropic diffusion. It is interesting to notice the equality

$$Q_c \operatorname{diag}(w_c(E_c))Q_c^* = Q \operatorname{diag}(w(E))Q^*, \tag{5.1}$$

which shows two alternative equivalent formulas for the graph diffusion operator.

For simplicity of presentation, we assume in the rest of the paper that the weights $w_c$ take only the values zero and one. Under this assumption, we introduce the matrix $P = \operatorname{diag}(w_c(E_c))$ and notice that $P$ is the matrix of an orthogonal projector on a subspace spanned by coordinate vectors with indices corresponding to the indices of edges present in $E$, and that equality (5.1) turns into

$$Q_c P Q_c^* = QQ^* = L. \tag{5.2}$$

Let us note that our results can be easily extended to a more general case of arbitrary nonnegative weights, or even to the case of the edge matrix $W$, assuming that it is symmetric nonnegative definite, $W = W^* \geq 0$.

Fiedler's pioneering work [8] on using the eigenpairs of the graph Laplacian to determine some structural properties of the graph has attracted much attention in the past. Recent advances in large-scale eigenvalue computations using multilevel preconditioning (e.g., [17], [20], [22]) suggest novel efficient numerical methods to compute the Fiedler vector and may rejuvenate this classical approach, e.g., for graph partitioning. In this paper, we concentrate on the whole set of eigenvalues of $L$, which is called the Laplacian graph spectrum.

It is known that the Laplacian graph spectrum does not determine the graph uniquely, i.e., that there exist isospectral graphs; see, e.g., [39] and references there. However, intuition suggests that a small change in a large graph should not change the Laplacian graph spectrum very much, and attempts have been made to use the closeness of Laplacian graph spectra to judge the closeness of the graphs in applications; for alternative approaches, see [2]. The goal of this section is to back up this intuition with rigorous estimates for proximity of the Laplacian graph spectra.

**5.2. Laplacian graph spectrum and Ritz values.** In the previous section, we obtained in Theorem 4.2 a weak majorization bound for changes in Ritz values depending on a change in the trial subspace, which we would like to apply to analyze the graph spectrum. In this subsection, we present an approach that allows us to interpret the Laplacian graph spectrum as a set of Ritz values obtained by the Rayleigh–Ritz method applied to the complete graph.

A graph $(V, E)$ can evidently be obtained from the complete graph $(V, E_c)$ by removing edges; moreover, as we already discussed, we can construct the $(V, E)$ graph Laplacian by either of the terms in equality (5.2). The problem is that such a construction cannot be recast as an application of the Rayleigh–Ritz method, since the multiplication by the projector $P$ takes place inside of the product in (5.2), not outside, as required by the Rayleigh–Ritz method.

To resolve this difficulty, we use the matrix $K = Q^*Q$, which is sometimes called the matrix of the graph [illegible] Laplacian, instead of the matrix of the graph [illegible] Laplacian $L = QQ^*$, as both matrices $K$ and $L$ share the same nonzero eigenvalues. The advantage of the edge Laplacian $K$ is that it can be obtained from the edge Laplacian of the complete graph $Q_c^*Q_c$ simply by removing the rows and columns that correspond to missing edges. Mathematically, this procedure can be viewed as an instance of the classical Rayleigh–Ritz method, as follows.

LEMMA 5.1. [illegible] $w_c$ [illegible] $P = \mathrm{diag}(w_c(E_c))$ [illegible] $E$ [illegible] $Q^*Q = (PQ_c^*Q_c)|_{\mathrm{Range}(P)}$. [illegible] $Q^*Q$ [illegible] $Q_c^*Q_c$ [illegible] $\mathrm{Range}(P)$

The application of the Rayleigh–Ritz method in this case is reduced to simply crossing out rows and columns of the matrix $Q_c^*Q_c$ corresponding to absent edges, since $P$ projects onto a span of coordinate vectors with the indices of the present edges.

Lemma 5.1 is a standard tool in spectral graph theory (see, e.g., [11]) for proving that the eigenvalues are interlacing; however, the procedure is not apparently recognized in the spectral graph community as an instance of the classical Rayleigh–Ritz method. Lemma 5.1 provides us with the missing link in order to apply our Theorem 4.2 to Laplacian graph spectra comparison.

**5.3. Majorization of Ritz values for Laplacian graph spectra comparison.** Using the tools that we have presented in the previous subsections, we can now apply the particular case, (4.4), of our weak majorization result of section 4 to analyze the change in the graph spectrum when several edges are added to or removed from the graph.

THEOREM 5.2. [illegible] $(V, E^1)$ [illegible] $(V, E^2)$ [illegible] $n$ [illegible] $V$ [illegible] $n_e$ [illegible] $E^1$ [illegible] $E^2$, [illegible] $E^1$ [illegible] $E^2$ [illegible] $l$ [illegible]

$$\sum_{k=1}^{n} |\lambda_k^1 - \lambda_k^2| \leq nl, \tag{5.3}$$

[illegible] $\lambda_k^1$ [illegible] $\lambda_k^2$ [illegible] $(V, E^1)$ [illegible] $(V, E^2)$ [illegible]

[illegible] The spectra of the graph vertex $n \times n$ Laplacian $QQ^*$ and the graph edge $n_e \times n_e$ Laplacian $Q^*Q$ are the same apart from zero, which does not affect the statement of the theorem, so we redefine $\lambda_k^1$ and $\lambda_k^2$ as elements of the spectra, counting the multiplicities, of the edge Laplacians of the graphs $(V, E^1)$ and $(V, E^2)$. Then, by Theorem 5.1, $\lambda_k^1$ and $\lambda_k^2$ are the Ritz values of the edge Laplacian matrix $A = Q_c^*Q_c$ of the complete graph, corresponding to the trial subspaces $\mathcal{X} = \mathrm{Range}(P_1)$ and $\mathcal{Y} = \mathrm{Range}(P_2)$ spanned by coordinate vectors with indices of the edges present in $E^1$ and $E^2$, respectively.

Let us apply Theorem 4.2, taking the sum over all available nonzero values in the weak majorization statement as in (4.4). This already gives us the left-hand side of (5.3). To obtain the right-hand side of (5.3) from Theorem 4.2, we now show in our case that, first, $\lambda_{\max} - \lambda_{\min} = n$ and, second, the sum of sines of all angles between the trial subspaces $\mathcal{X}$ and $\mathcal{Y}$ is equal to $l$.

The first claim follows from the fact, which is easy to check by direct calculation, that the spectrum of the vertex (and thus the edge) Laplacian of the complete graph with $n$ vertices consists of only two eigenvalues $\lambda_{\max} = n$ and $\lambda_{\min} = 0$. Let us make a side note that we can interpret the Laplacian of the complete graph as a scaled projector; i.e., in this case we could have applied Theorem 3.3 directly, rather than Theorem 4.2, which would still result in (5.3).

The second claim, on the sum of sines of all angles, follows from the definitions of $\mathcal{X}$ and $\mathcal{Y}$ and the assumption that the number of differing edges in $E^1$ and $E^2$ is equal to $l$. Indeed, $\mathcal{X}$ and $\mathcal{Y}$ are spanned by coordinate vectors with indices of the edges present in $E^1$ and $E^2$. The edges that are present in both $E^1$ and $E^2$ contribute zero angles into $\Theta(\mathcal{X}, \mathcal{Y})$, while the $l$ edges that are different in $E^1$ and $E^2$ contribute $l$ right angles into $\Theta(\mathcal{X}, \mathcal{Y})$, so that the sum of all terms in $\sin \Theta(\mathcal{X}, \mathcal{Y})$ is equal to $l$. $\square$

Remark 4.1 is also applicable for Theorem 5.2—while the min term is always zero, since all graph Laplacians are degenerate, the max term can be made smaller by replacing $n$ with the largest eigenvalue of the Laplacian of the graph $(V, E^1 \cup E^2)$.

It is clear from the proof that we do not use the full force of our weak majorization results in Theorem 5.2, because it concerns angles which are zero or $\pi/2$. Nevertheless, the results of Theorem 5.2 appear to be novel in graph theory. We note that these results can easily be extended on [illegible]-partite graphs, and possibly to mixed graphs.

Let us finally mention an alternative approach to compare Laplacian graph spectra, which we do not cover in the present paper, by applying Corollary 2.5 directly to graph Laplacians and estimating the right-hand side, using the fact that the changes in $l$ edges represent a low-rank perturbation of the graph Laplacian; cf. [36].

## 6. Conclusions.

We use majorization to investigate the sensitivity of angles between subspaces and Ritz values with respect to subspaces, and to analyze changes in graph Laplacian spectra where edges are added and removed. We discover that these seemingly different areas are all surprisingly related. We establish in a unified way new results on weak majorization of the changes in the sine/cosine (squared) and in the Ritz values. The main strength of the paper in our opinion is, however, not so much in the results themselves but rather in a novel and elegant proof technique that is based on a classical but rarely used idea of extending Hermitian operators to orthogonal projectors in a larger space. We believe that this technique is very powerful and should be known to a wider audience.

**Acknowledgments.** We are indebted to Li Qiu and Yanxia Zhang, who suggested to us the idea of the proof of the sine estimate in Theorem 3.2, which also allowed us to simplify our original proof of Theorem 3.3. We also thank our Ph.D. students Ilya Lashuk, for contributing to section 2.3, and Abram Jujunashvili, for many helpful comments. Finally, we thank the anonymous referees for their numerous useful and illuminating suggestions, which dramatically improved the paper.

## REFERENCES

[1] R. Bhatia, *Matrix Analysis*, Springer-Verlag, Berlin, 1997.

[2] V. D. Blondel, A. Gajardo, M. Heymans, P. Senellart, and P. Van Dooren, *A measure of similarity between graph vertices: Applications to synonym extraction and web searching*, SIAM Rev., 46 (2004), pp. 647–666.

[3] F. Chatelin, *Eigenvalues of Matrices*, John Wiley and Sons, Chichester, UK, 1993.

[4] F. R. K. Chung, *Spectral Graph Theory*, CBMS Reg. Conf. Ser. Math. 92,. AMS, Providence, RI, 1997.

[5] D. M. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs: Theory and Applications*, 3rd ed., Johann Ambrosius Barth, Heidelberg, Germany, 1995.

[6] C. Davis, *Separation of two linear subspaces*, Acta Sci. Math. (Szeged), 19 (1958), pp. 172–187.

[7] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation*, III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

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

[9] I. M. Glazman and Ju. I. Ljubič, *Finite-dimensional Linear Analysis: A Systematic Presentation in Problem Form*, translated from the Russian and edited by G. P. Barker and G. Kuerti, M.I.T. Press, Cambridge, MA, 1974.

[10] I. C. Gohberg and M. G. Kreĭn, *Introduction to the Theory of Linear Nonselfadjoint Operators*, translated from the Russian by A. Feinstein, Trans. Math. Monogr. 18, AMS, Providence, RI, 1969.

[11] W. H. Haemers, *Interlacing eigenvalues and graphs*, Linear Algebra Appl., 226/228 (1995), pp. 593–616.

[12] P. R. Halmos, *Normal dilations and extensions of operators*, Summa Brasil. Math., 2 (1950), pp. 125–134.

[13] P. R. Halmos, *Two subspaces*, Trans. Amer. Math. Soc., 144 (1969), pp. 381–389.

[14] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, Cambridge, UK, 1959.

[15] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1999.

[16] C. Jordan, *Essai sur la géométrie à n dimensions*, Bull. Soc. Math. France, 3 (1875), pp. 103–174.

[17] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[18] A. V. Knyazev and M. E. Argentati, *Principal angles between subspaces in an A-based scalar product: Algorithms and perturbation estimates*, SIAM J. Sci. Comput., 23 (2002), pp. 2009–2040.

[19] A. V. Knyazev and M. E. Argentati, *On proximity of Rayleigh quotients for different vectors and Ritz values generated by different trial subspaces*, Linear Algebra Appl., 415 (2006), pp. 82–95.

[20] A. V. Knyazev and K. Neymeyr, *Efficient solution of symmetric eigenvalue problems using multigrid preconditioners in the locally optimal block conjugate gradient method*, Electron. Trans. Numer. Anal., 15 (2001), pp. 38–55.

[21] A. V. Knyazev and J. E. Osborn, *New a priori FEM error estimates for eigenvalues*, SIAM J. Numer. Anal., 43 (2006), pp. 2647–2667; extended online version available at http://www-math.cudenver.edu/ccm/reports/rep215.pdf.

[22] Y. Koren, L. Carmel, and D. Harel, *Drawing huge graphs by algebraic multigrid optimization*, Multiscale Model. Simul., 1 (2003), pp. 645–673.

[23] S. Kosinov and T. Caelli, *Inexact multisubgraph matching using graph eigenspace and clustering models*, in Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, Windsor, ON, 2002, Lecture Notes in Comput. Sci. 2396, Springer-Verlag, New York, pp. 133–142.

[24] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Math. Sci. Engrg. 143, Academic Press, New York, 1979.

[25] P. McDonald and R. Meyers, *Diffusions on graphs, Poisson problems and spectral geometry*, Trans. Amer. Math. Soc., 354 (2002), pp. 5111–5136.

[26] R. Merris, *Laplacian matrices of graphs: A survey*, Linear Algebra Appl., 197/198 (1994), pp. 143–176.

[27] R. Merris, *A survey of graph Laplacians*, Linear and Multilinear Algebra, 39 (1995), pp. 19–31.

[28] R. Merris, *Laplacian graph eigenvectors*, Linear Algebra Appl., 278 (1998), pp. 221–236.

[29] B. Mohar, *Some applications of Laplace eigenvalues of graphs*, in Graph Symmetry (Montreal, PQ, 1996), NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci. 497, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 225–275.

[30] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[31] C. C. Paige and M. Wei, *History and generality of the CS decomposition*, Linear Algebra Appl., 208/209 (1994), pp. 303–326.

[32] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Appl. Math. 20, SIAM, Philadelphia, 1997.

[33] L. Qiu and Y. Zhang, *private communication*, 2005.

[34] L. Qiu, Y. Zhang, and C.-K. Li, *Unitarily invariant metrics on the Grassmann space*, SIAM

J. Matrix Anal. Appl., 27 (2005), pp. 507–531.
[35] F. Riesz and B. Sz.-Nagy, *Functional Analysis*, Dover Publications, New York, 1990.
[36] W. So, *Rank one perturbation and its application to the Laplacian spectrum of a graph*, Linear and Multilinear Algebra, 46 (1999), pp. 193–198.
[37] G. W. Stewart and J. G. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, MA, 1990.
[38] G. W. Stewart, *Matrix Algorithms Volume* II: *Eigensystems*, SIAM, Philadelphia, 2001.
[39] E. R. van Dam and W. H. Haemers, *Which graphs are determined by their spectrum*? Linear Algebra Appl., 373 (2003), pp. 241–272.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 33–53

# SOLUTIONS OF THE PARTIALLY DESCRIBED INVERSE QUADRATIC EIGENVALUE PROBLEM*

YUEN-CHENG KUO[†], WEN-WEI LIN[‡], AND SHU-FANG XU[§]

**Abstract.** Given $k$ pairs of complex numbers and vectors (closed under conjugation), we consider the inverse quadratic eigenvalue problem of constructing $n \times n$ real symmetric matrices $M$, $C$, and $K$ (with $M$ positive definite) so that the quadratic pencil $Q(\lambda) \equiv \lambda^2 M + \lambda C + K$ has the given $k$ pairs as eigenpairs. Using various matrix decompositions, we first construct a general solution to this problem with $k \leq n$. Then, with appropriate choices of degrees of freedom in the general solution, we construct several particular solutions with additional eigeninformation or special properties. Numerical results illustrating these solutions are also presented.

**Key words.** quadratic eigenvalue problem, inverse eigenvalue problem, partially prescribed spectrum, partial eigenstructure assignment

**AMS subject classifications.** 65F15, 15A22, 65H17, 93B55

**DOI.** 10.1137/05064134X

## 1. Introduction.

This paper first constructs a general symmetric quadratic pencil

$$Q(\lambda) \equiv \lambda^2 M + \lambda C + K \tag{1.1}$$

with $M^\top = M > 0$ (being symmetric positive definite), $C^\top = C$, and $K^\top = K \in \mathbb{R}^{n\times n}$, so that $Q(\lambda)$ has $k$ given pairs of complex numbers and vectors, closed under conjugation, as its eigenpairs. Then, under appropriate choices of degrees of freedom in the general solution, we construct several particular solutions with additional eigeninformation or special properties. Here, we formulate our partially described inverse quadratic eigenvalue problem.

PD-IQEP (partially described inverse quadratic eigenvalue problem). [illegible] $(\Lambda, X) \in \mathbb{R}^{k\times k} \times \mathbb{R}^{n\times k}$ $(k \leq n)$ [illegible]

$$\Lambda = \operatorname{diag}\{\lambda_1^{[2]}, \ldots, \lambda_\ell^{[2]}; \lambda_{2\ell+1}, \ldots, \lambda_k\} \tag{1.2a}$$

[illegible]

$$\lambda_j^{[2]} = \begin{bmatrix} \alpha_j & \beta_j \\ -\beta_j & \alpha_j \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \beta_j \neq 0, \quad [illegible] \quad j = 1, \ldots, \ell, \tag{1.2b}$$

[illegible]

$$X = [\mathbf{x}_{1R}, \mathbf{x}_{1I}, \ldots, \mathbf{x}_{\ell R}, \mathbf{x}_{\ell I}; \mathbf{x}_{2\ell+1}, \ldots, \mathbf{x}_k], \tag{1.3}$$

---

[*] Received by the editors September 28, 2005; accepted for publication (in revised form) by M. Chu May 8, 2006; published electronically December 21, 2006.
http://www.siam.org/journals/simax/29-1/64134.html

[†] Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung, 811, Taiwan (yckuo@nuk.edu.tw).

[‡] Department of Mathematics, National Tsinghua University, Hsinchu, 300, Taiwan (wwlin@am.nthu.edu.tw).

[§] LMAM, School of Mathematical Sciences, Peking University, Beijing, 100871, China (xsf@pku.edu.cn).

[illegible] $n \times n$ [illegible] $M$ [illegible] $C$ [illegible] $K$ [illegible] $M$ [illegible]

$$MX\Lambda^2 + CX\Lambda + KX = 0. \tag{1.4}$$

This problem is called "partially described" because the quadratic pencil (1.1) has part of its eigenvalues and the corresponding eigenvectors, respectively, given by

$$\begin{aligned} &\alpha_1 \pm \boldsymbol{\iota}\beta_1, \ldots, \alpha_\ell \pm \boldsymbol{\iota}\beta_\ell;\ \lambda_{2\ell+1}, \ldots, \lambda_k; \\ \text{and}\ &\mathbf{x}_{1R} \pm \boldsymbol{\iota}\mathbf{x}_{1I}, \ldots, \mathbf{x}_{\ell R} \pm \boldsymbol{\iota}\mathbf{x}_{\ell I};\ \mathbf{x}_{2\ell+1}, \ldots, \mathbf{x}_k. \end{aligned} \tag{1.5}$$

Here $\boldsymbol{\iota} = \sqrt{-1}$. We note that in a large and complicated physical system [16, 17, 19], it is often impractical or impossible to obtain complete spectral information. Thus, it is more sensible to consider a PD-IQEP where only a subset of eigenpairs is known.

In mathematical modeling, there is often a correspondence between the internal parameters and the external behavior. Finding the eigenpairs $(\lambda, \mathbf{x})$ such that $Q(\lambda)\mathbf{x} = 0$ for given $M$, $C$, and $K$ is referred to as a direct quadratic eigenvalue problem (QEP). This is part of the process to induce the dynamics behavior of a system from known physical parameters such as mass, length, elasticity, inductance, capacitance, and so on. A detailed theoretical analysis of QEPs can be found in [10]. Engineering applications, mathematical properties, and a variety of numerical methods for QEPs can be found in the recent survey paper [18]. In contrast, the inverse problem is to determine or estimate some parameters of the system from its measured or expected behavior. The concern in the direct problem is to deduce the behavior from the parameters, whereas in the inverse problem we try to recover the parameters from the behavior. The inverse problem is as important as the direct problem in application.

There is much interest in the inverse eigenvalue problem, including the pole assignment problem. Some general reviews and extensive bibliographies can be found in [5, 3]. Some previous attempts at solving the IQEP are listed as follows:

(i) In [4], special symmetric solutions $M$, $C$, and $K$ (with $M$ and $K$ being symmetric positive definite) to the standard PD-IQEP (with $k \leq n$) and the monic PD-IQEP (with $k = n + 1$) have been constructed.

(ii) In [12], symmetric solutions $M$, $C$, and $K$ (with $M$ and $K$ being symmetric positive definite, and $C$ being positive semidefinite) have been constructed when all eigenvalues are simple and nonreal, and the corresponding eigenvector matrix is of the form $X = X_R(I - \boldsymbol{\iota}\Theta)$, where $X_R$ is nonsingular and $\Theta$ is orthogonal.

(iii) In [2, 7, 6, 14, 15], a feedback control with partial eigenstructure assignment was considered. The proportional and derivative feedback controllers have been constructed to assign specific eigenpairs to the new QEP and make the closed-loop system insensitive to perturbation. However, this consideration cannot preserve the symmetry of the closed-loop system.

(iv) In [1, 13], a symmetric $Q(\lambda)$ has been constructed to partially assign some eigenvalues while retaining other eigenpairs.

(v) In [8, 9, 11], for the finite element model updating problem, a symmetric $Q(\lambda)$ which possesses partially described eigenpairs and is nearest to the original analytical model has been constructed.

(vi) Other types of IQEPs have been studied under modified conditions. For instance, in [15], a symmetric quadratic pencil $Q(\lambda) = \lambda^2 I + \lambda C + K$ has been found, so that $Q(\lambda)$ and $\hat{Q}(\lambda)$ (constructed from $Q(\lambda)$ by deleting the last row and column) have prescribed eigenvalues. In [17], nonproportional underdamped systems have been studied.

The main purpose of this paper is first to construct a general solution of the PD-IQEP. With appropriate choices of free variables in the general solution, we then construct several particular solutions to the PD-IQEP with additional eigeninformation or special structures. The particular solutions of the standard PD-IQEP and the monic PD-IQEP in [4] are just the special cases of the general solution (see section 5).

This paper is organized as follows. In section 2, we give an expression of the general solution to the PD-IQEP in terms of decompositions of some associated matrices. In section 3, we construct particular solutions with $K \geq 0$. In section 4, with $k = n$, we construct particular solutions assigning additional eigenvalues or eigenpairs. In section 5, with $k < n$, we construct particular solutions assigning one additional complex eigenpair or special structures. Numerical results, illustrating the particular solutions in section 4, are presented in section 6. A conclusion and a list of solved and unsolved PD-IQEPs are presented in section 7.

Throughout this paper, we use capital letters to denote matrices, and lowercase (bold) letters to denote scalars (vectors). For $B \in \mathbb{R}^{n\times m}$, $B^\top$, $\bar{B}$, and $B^H$ denote the transpose, conjugate, and conjugate transpose of $B$, respectively. $\mathcal{N}(B)$ denotes the null space of $B$. For a symmetric $A \in \mathbb{R}^{n\times n}$, $A > 0$ ($\geq 0$) denotes a symmetric positive definite (semidefinite) matrix. The spectrum and the spectral radius of $A$ are denoted by $\sigma(A)$ and $\rho(A)$, respectively.

For simplicity, we make the following assumptions:

(H1) The eigenvector matrix $X$ in (1.3) has full column rank, i.e., $\operatorname{rank}(X) = k$.

(H2) The eigenvalue matrix $\Lambda$ in (1.2a) has only simple eigenvalues.

**2. General solution of the PD-IQEP.** In this section, we shall solve a general solution to the PD-IQEP for a given matrix pair $(\Lambda, X) \in \mathbb{R}^{k\times k} \times \mathbb{R}^{n\times k}$ ($k \leq n$) as in (1.2) and (1.3).

THEOREM 2.1. [illegible] $(\Lambda, X) \in \mathbb{R}^{k\times k} \times \mathbb{R}^{n\times k}$ $(k \leq n)$ [illegible] (1.2) [illegible] (1.3) [illegible]

$$X = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{2.1}$$

[illegible] $X$ [illegible] $Q \in \mathbb{R}^{n\times n}$ [illegible] $R \in \mathbb{R}^{k\times k}$ [illegible] $S = R\Lambda R^{-1}$ [illegible] **PD-IQEP** [illegible]

$$M = Q \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} Q^\top,\ C = Q \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} Q^\top,\ K = Q \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} Q^\top, \tag{2.2}$$

[illegible]

(i) $\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{R}^{n\times n}$ [illegible],

(ii) $C_{22} = C_{22}^\top, K_{22} = K_{22}^\top \in \mathbb{R}^{(n-k)\times(n-k)}$ [illegible],

(iii) $C_{21} = C_{12}^\top \in \mathbb{R}^{(n-k)\times k}$ [illegible],

(2.3a) (iv) $C_{11} = C_{11}^\top = -(M_{11}S + S^\top M_{11} + R^{-\top} D R^{-1}) \in \mathbb{R}^{k\times k}$,

(2.3b) (v) $K_{11} = K_{11}^\top = S^\top M_{11} S + R^{-\top} D \Lambda R^{-1} \in \mathbb{R}^{k\times k}$,

(2.3c) (vi) $K_{21} = K_{12}^\top = -(M_{21}S^2 + C_{21}S) \in \mathbb{R}^{(n-k)\times k}$,

[illegible]

$$D = \operatorname{diag}\left( \begin{bmatrix} \xi_1 & \eta_1 \\ \eta_1 & -\xi_1 \end{bmatrix}, \ldots, \begin{bmatrix} \xi_\ell & \eta_\ell \\ \eta_\ell & -\xi_\ell \end{bmatrix}; \ \xi_{2\ell+1}, \ldots, \xi_k \right), \tag{2.4}$$

[illegible] $\xi_i$ [illegible] $\eta_i$ [illegible]. Substituting (2.1) and (2.2) into (1.4), we have

$$M_{11}S^2 + C_{11}S + K_{11} = 0, \tag{2.5}$$

$$M_{21}S^2 + C_{21}S + K_{21} = 0. \tag{2.6}$$

Thus, finding $M$, $C$, and $K$ which satisfy (1.4) is equivalent to finding the submatrices $M_{11}$, $M_{21}$, $C_{11}$, $C_{21}$, $K_{11}$, and $K_{21}$ which satisfy (2.5) and (2.6). Clearly, it follows from (2.6) that $K_{21}$ is determined by (2.3c), where $M_{21}$ and $C_{21}$ are arbitrary.

As $M$ and $K$ are required to be symmetric positive definite and symmetric, respectively, so are $M_{11}$ and $K_{11}$ in (2.2). From (2.5) it follows that

$$K_{11} = -(M_{11}S^2 + C_{11}S). \tag{2.7}$$

Let $M_{11}$ be an arbitrary symmetric positive definite matrix. We need to find a symmetric $C_{11}$ such that $K_{11}$ in (2.7) is symmetric. We thus need a $C_{11} = C_{11}^\top$ so that

$$(M_{11}S^2 + C_{11}S)^\top = M_{11}S^2 + C_{11}S. \tag{2.8}$$

After rearrangement, (2.8) becomes

$$C_{11}S - S^\top C_{11} = -M_{11}S^2 + (S^2)^\top M_{11}. \tag{2.9}$$

It is easily seen that (2.9) has a particular solution

$$C_{11}^0 = -(M_{11}S + S^\top M_{11}). \tag{2.10}$$

Next we consider of the homogeneous equation

$$C_{11}S - S^\top C_{11} = 0. \tag{2.11}$$

Substituting $S = R\Lambda R^{-1}$ into (2.11), we get

$$(R^\top C_{11}R)\Lambda - \Lambda^\top (R^\top C_{11}R) = 0. \tag{2.12}$$

Partitioning $R^\top C_{11}R$ compatibly with $\Lambda$, we have $s = k - \ell$ and

$$R^\top C_{11} R = \begin{bmatrix} \Gamma_{11} & \cdots & \Gamma_{1s} \\ \vdots & \ddots & \vdots \\ \Gamma_{s1} & \cdots & \Gamma_{ss} \end{bmatrix}, \tag{2.13}$$

where $\Gamma_{jj}$ is a $2 \times 2$ matrix for $1 \le j \le \ell$ and $\Gamma_{jj}$ is a $1 \times 1$ matrix for $\ell + 1 \le j \le s$. Substituting (2.13) into (2.12) and using assumption (H2), we deduce that

$$\begin{gathered} \Gamma_{ji} = 0, \qquad j \neq i, \\ \Gamma_{jj}\lambda_j^{[2]} - (\lambda_j^{[2]})^\top \Gamma_{jj} = 0, \qquad j = 1, \ldots, \ell; \end{gathered} \tag{2.14}$$

and

$$\Gamma_{\ell+j,\ell+j}\lambda_{2\ell+j} - \lambda_{2\ell+j}\Gamma_{\ell+j,\ell+j} = 0, \qquad j = 1, \ldots, s-\ell. \tag{2.15}$$

Since $\lambda_j^{[2]}$ has the form in (1.2b) with $\beta_j \neq 0$, it is easily seen that the general solution of (2.14) has the form

$$\Gamma_{jj} = \begin{bmatrix} \xi_j & \eta_j \\ \eta_j & -\xi_j \end{bmatrix}, \quad j = 1, \ldots, \ell, \tag{2.16}$$

where $\xi_j$, $\eta_j$ are arbitrary real numbers and (2.15) holds for any real numbers $\Gamma_{\ell+j,\ell+j} = \xi_{\ell+j}$. Thus, the general solution of the homogeneous equation (2.11) has the form

$$C_{11} = R^{-\top} D R^{-1}, \tag{2.17}$$

with $D$ defined in (2.4). This, together with (2.10), gives rise to the general solution of (2.9):

$$C_{11} = -R^{-\top} D R^{-1} - M_{11}S - S^{\top}M_{11}, \tag{2.18}$$

(cf. (2.3a)). Substituting (2.18) into (2.7) yields (2.3b). Note that from (2.16) and (1.2) the matrix $D\Lambda$ in (2.3b) is symmetric. This completes the proof. ☐

[illegible] 2.1. Theorem 2.1 shows that the solution to the PD-IQEP is underdetermined. Therefore, the question arises as to how these degrees of freedom could be exploited. This will be discussed in the subsequent sections.

## 3. Particular solutions with $K \geq 0$.

In practice, the matrix $K$ in the PD-IQEP is sometimes required to be symmetric positive semidefinite. In this section, we shall apply Theorem 2.1 to construct such a solution. We first prove the following two lemmas.

LEMMA 3.1. [illegible] $D$ [illegible] (2.4) [illegible] $M_{11}$ [illegible] $K_{11}$ [illegible] (2.3b) [illegible]

[illegible]. Since $S = R\Lambda R^{-1}$, it is easy to see that $K_{11}$ in (2.3b) is symmetric positive semidefinite if and only if the matrix

$$\Lambda^{\top} R^{\top} M_{11} R\Lambda + D\Lambda \tag{3.1}$$

is symmetric positive semidefinite.

Since $\Lambda$ has distinct eigenvalues, we have either $0 \notin \sigma(\Lambda)$ or 0 being a simple eigenvalue of $\Lambda$ (say, $\lambda_k = 0$). We first construct a symmetric positive definite (or a symmetric positive semidefinite when $\lambda_k = 0$) matrix $\widetilde{M}$ so that $\widetilde{M} + D\Lambda \geq 0$. Then we use $\widetilde{M}$ to construct the desired $M_{11}$.

Take

$$\widetilde{M} = \begin{cases} \widetilde{M}_1 & \text{if } 0 \notin \sigma(\Lambda), \\ \operatorname{diag}(\widetilde{M}_1,\ 0) & \text{if } 0 \in \sigma(\Lambda), \end{cases} \tag{3.2}$$

where

$$\widetilde{M}_1 = \operatorname{diag}\left( \begin{bmatrix} x_1 & z_1 \\ z_1 & y_1 \end{bmatrix}, \ldots, \begin{bmatrix} x_\ell & z_\ell \\ z_\ell & y_\ell \end{bmatrix};\ x_{2\ell+1}, \ldots, x_{k_1} \right), \tag{3.3}$$

in which $k_1 = k$ when $0 \notin \sigma(\Lambda)$ and $k_1 = k-1$ when $\lambda_k = 0$. From (2.4) and (1.2), we denote

$$D\Lambda = \operatorname{diag}\left( \begin{bmatrix} \theta_1 & \omega_1 \\ \omega_1 & -\theta_1 \end{bmatrix}, \ldots, \begin{bmatrix} \theta_\ell & \omega_\ell \\ \omega_\ell & -\theta_\ell \end{bmatrix};\ \theta_{2\ell+1}, \ldots, \theta_k \right), \tag{3.4}$$

where

$$\begin{array}{l} \theta_j = \alpha_j\xi_j - \beta_j\eta_j, \quad \omega_j = \alpha_j\eta_j + \beta_j\xi_j, \quad j = 1, \ldots, \ell, \\ \theta_j = \xi_j\lambda_j, \quad j = 2\ell+1, \ldots, k, \end{array} \tag{3.5}$$

with $\xi_j$ and $\eta_j$ being arbitrary real numbers. Using (3.4) and (3.5), if we choose $x_i$, $y_i$, and $z_i$ such that

$$\begin{cases} x_i > 0, & i = 1, \ldots, k_1, \\ x_i y_i - z_i^2 > 0, & i = 1, \ldots, \ell, \end{cases} \tag{3.6}$$

$$x_i + \theta_i \geq 0, \qquad i = 2\ell+1, \ldots, k_1, \tag{3.7}$$

$$\begin{cases} x_i + \theta_i > 0, \quad d_i \equiv (y_i - \theta_i) - \frac{(z_i+\omega_i)^2}{x_i+\theta_i} \geq 0, \\ \text{or} \quad x_i + \theta_i = z_i + \omega_i = 0, \quad y_i - \theta_i \geq 0, \end{cases} \qquad i = 1, \ldots, \ell, \tag{3.8}$$

then $\widetilde{M}_1 > 0$ and $\widetilde{M} + D\Lambda \geq 0$. Obviously, such real numbers $x_i$, $y_i$, and $z_i$ can be easily chosen. Once $\widetilde{M}_1$ is determined, the required $M_{11}$ can be chosen by

$$M_{11} = \begin{cases} R^{-\top}\Lambda^{-\top}\widetilde{M}_1\Lambda^{-1}R^{-1} & \text{if} \quad 0 \notin \sigma(\Lambda), \\ R^{-\top} \begin{bmatrix} \Lambda_1^{-\top}\widetilde{M}_1\Lambda_1^{-1} & 0 \\ 0 & 1 \end{bmatrix} R^{-1} & \text{if} \quad 0 \in \sigma(\Lambda), \end{cases} \tag{3.9}$$

where $\Lambda_1 = \Lambda(1:k-1, 1:k-1)$. $\square$

Let $K_{11}$ be constructed as in Lemma 3.1. Then from (3.2) and (3.9) it is easily seen that

$$K_{11} = R^{-\top}(\widetilde{M} + D\Lambda)R^{-1}. \tag{3.10}$$

Since $\widetilde{M} + D\Lambda \geq 0$, there exists a congruence transformation $L$ such that

$$L^\top(\widetilde{M} + D\Lambda)L = \begin{bmatrix} \widetilde{K}_1 & 0 \\ 0 & 0_q \end{bmatrix}, \tag{3.11}$$

where $\widetilde{K}_1$ is a $(k-q) \times (k-q)$ positive diagonal matrix. Since $K$ is required to be symmetric positive semidefinite, it follows that

$$\begin{bmatrix} RL & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} RL & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} L^T R^T K_{11} RL & L^T R^T K_{12} \\ K_{21} RL & K_{22} \end{bmatrix} \geq 0.$$

From (3.10) and (3.11), it holds that $L^\top R^\top K_{11} RL$ has the form in (3.11), and then the last $q$ columns of $K_{21}RL$ must be zero. In the following lemma we show that $M_{21}$ and $C_{21}$ can be chosen so that this condition holds.

LEMMA 3.2. [illegible] $k \times k$ [illegible] $L$ [illegible] (3.11) [illegible] $M_{21}$ [illegible] $C_{21}$ [illegible] $q$ [illegible] $K_{21}RL$ [illegible] $K_{21}$ [illegible] (2.3c) [illegible]. We first consider the case when $0 \notin \sigma(\Lambda)$. In this case, since $R\Lambda^2 L$ and $R\Lambda L$ are nonsingular, there exist orthogonal matrices $Q_1$ and $Q_2$ such that

$$Q_1^\top R\Lambda^2 L \begin{bmatrix} 0 \\ I_q \end{bmatrix} = \begin{bmatrix} \Gamma_1 \\ 0 \end{bmatrix}, \quad Q_2^\top R\Lambda L \begin{bmatrix} 0 \\ I_q \end{bmatrix} = \begin{bmatrix} \Delta_1 \\ 0 \end{bmatrix}, \tag{3.12}$$

where $\Gamma_1, \Delta_1 \in \mathbb{R}^{q\times q}$ are nonsingular. Now let

$$M_{21}Q_1 = [M_{21}^1 \mid M_{21}^2], \qquad C_{21}Q_2 = [C_{21}^1 \mid C_{21}^2], \tag{3.13}$$

where $M_{21}^1,\ C_{21}^1 \in \mathbb{R}^{k\times q}$. It follows from (2.3c), (3.12), and (3.13) that

$$\begin{aligned} 0 &= K_{21}RL \begin{bmatrix} 0 \\ I_q \end{bmatrix} && (3.14) \\ &= -(M_{21}S^2 + C_{21}S)RL \begin{bmatrix} 0 \\ I_q \end{bmatrix} \\ &= [M_{21}^1 \mid M_{21}^2] \begin{bmatrix} \Gamma_1 \\ 0 \end{bmatrix} + [C_{21}^1 \mid C_{21}^2] \begin{bmatrix} \Delta_1 \\ 0 \end{bmatrix}. && (3.15) \end{aligned}$$

Therefore, with $C_{21}^1$ being arbitrary,

$$M_{21}^1 = -C_{21}^1 \Delta_1 \Gamma_1^{-1}. \tag{3.16}$$

Similarly, in the case when $0 \in \sigma(\Lambda)$, there exist orthogonal matrices $Q_1$ and $Q_2$ such that

$$Q_1^\top R\Lambda^2 L \begin{bmatrix} 0 \\ I_q \end{bmatrix} = \begin{bmatrix} \Gamma_1 & \boldsymbol{\gamma}_1 \\ 0 & 0 \end{bmatrix}, \qquad Q_2^\top R\Lambda L \begin{bmatrix} 0 \\ I_q \end{bmatrix} = \begin{bmatrix} \Delta_1 & \boldsymbol{\delta}_1 \\ 0 & 0 \end{bmatrix}, \tag{3.17}$$

where $\Gamma_1, \Delta_1 \in \mathbb{R}^{(q-1)\times(q-1)}$ are nonsingular and $\boldsymbol{\gamma}_1, \boldsymbol{\delta}_1 \in \mathbb{R}^{q-1}$. Let

$$M_{21}Q_1 = [M_{21}^1 \mid M_{21}^2], \qquad C_{21}Q_2 = [C_{21}^1 \mid C_{21}^2], \tag{3.18}$$

where $M_{21}^1,\ C_{21}^1 \in \mathbb{R}^{k\times(q-1)}$, and let

$$0 = K_{21}RL \begin{bmatrix} 0 \\ I_q \end{bmatrix} = -(M_{21}S^2 + C_{21}S)RL \begin{bmatrix} 0 \\ I_q \end{bmatrix}.$$

We have

$$M_{21}^1 = -C_{21}^1 \Delta_1 \Gamma_1^{-1}, \tag{3.19}$$

where $C_{21}^1$ solves $C_{21}^1(\boldsymbol{\delta}_1 - \Delta_1\Gamma_1^{-1}\boldsymbol{\gamma}_1) = 0$, which has infinite many solutions.

Thus, we have completed the proof of the lemma. □

Using Lemmas 3.1 and 3.2, we can construct a particular solution to the PD-IQEP with $K \geq 0$ as follows.

ALGORITHM 3.1. SOLVING THE PD-IQEP WITH $K \geq 0$.

[illegible] 1. Choose $D$ as in (2.4) arbitrarily and compute $D\Lambda$ by (3.4) and (3.5).

[illegible] 2. Construct a positive definite matrix $M_{11}$ by (3.2)–(3.9), compute $C_{11}$ and $K_{11}$ by (2.3a) and (2.3b), respectively, and compute $L$ as in (3.11).

[illegible] 3. Compute the decomposition (3.12) (or (3.17)), determine $M_{21}$ and $C_{21}$ by (3.13) and (3.16) (or (3.18) and (3.19)), and compute $K_{21}$ by (2.3c).

[illegible] 4. Choose an $(n-k)\times(n-k)$ positive matrix $\widehat{M}$ and an $(n-k)\times(n-k)$ semipositive matrix $\widehat{K}$ and compute $M_{22} = \widehat{M} + M_{21}M_{11}^{-1}M_{21}^T$ and $K_{22} = \widehat{K} + K_{21}K_{11}^{\dagger}K_{21}^T$. Here $K_{11}^{\dagger}$ denotes the Moore–Penrose inverse of $K_{11}$.

[illegible] 5. Choose an arbitrary symmetric $C_{22}$ and form

$$M = Q\begin{bmatrix} M_{11} & M_{21}^\top \\ M_{21} & M_{22} \end{bmatrix} Q^\top, \quad C = Q\begin{bmatrix} C_{11} & C_{21}^\top \\ C_{21} & C_{22} \end{bmatrix} Q^\top, \quad K = Q\begin{bmatrix} K_{11} & K_{21}^\top \\ K_{21} & K_{22} \end{bmatrix} Q^\top,$$

where $Q$ is given by (2.1).

**4. Particular solutions with additional eigeninformation when $k = n$.** Since there are still many degrees of freedom in the general solution of the PD-IQEP in section 3, we are motivated to satisfy additional constraints or eigeninformation so that the number of equations constructed from these additional conditions is less than or equal to the number of free variables. Consequently, such a PD-IQEP with the additional eigeninformation can be solved generically.

By Theorem 2.1, with $k = n$, the general solution of the PD-IQEP is given by

$$C = -(MS + S^\top M + R^{-\top}DR^{-1}), \tag{4.1a}$$

$$K = S^\top MS + R^{-\top}D\Lambda R^{-1}, \tag{4.1b}$$

where $M > 0$ can be arbitrarily chosen and $D$ is given by (2.4) with $k = n$. Based on the factorization of quadratic matrix pencils [10, p. 228], we then have the following theorem.

THEOREM 4.1. [illegible] $k = n$ [illegible]

$$Q(\lambda) = \lambda^2 M + \lambda C + K = (\lambda I - S^\top - R^{-\top}DR^{-1}M^{-1})M(\lambda I - S). \tag{4.2}$$

[illegible] $Q(\lambda)$ [illegible] $S$ ( [illegible] $\Lambda$) [illegible] $S + M^{-1}R^{-\top}DR^{-1}$

[illegible]. Substituting (4.1) into $Q(\lambda)$, we have

$$\begin{aligned} Q(\lambda) &= \lambda^2 M + \lambda C + K \\ &= \lambda^2 M - \lambda(MS + S^\top M + R^{-\top}DR^{-1}) + S^\top MS + R^{-\top}D\Lambda R^{-1} \\ &= (\lambda I - S)^\top M(\lambda I - S) + R^{-\top}DR^{-1}(-\lambda I + S) \\ &= \left((\lambda I - S)^\top M - R^{-\top}DR^{-1}\right)(\lambda I - S) \\ &= (\lambda I - S^\top - R^{-\top}DR^{-1}M^{-1})M(\lambda I - S). \qquad (4.3) \end{aligned}$$

Thus (4.2) holds. □

In the rest of this section, we shall present two particular solutions to the PD-IQEP with additional eigeninformation when $k = n$.

**4.1. Particular solutions with additional eigenvalues.** Our goal in this subsection is to construct a quadratic pencil $Q(\lambda)$ in (4.3) which has $n$ arbitrarily assigned eigenvalues, closed under complex conjugation, by appropriate choices of $M$ and $D$.

THEOREM 4.2. [illegible] $M$ [illegible] $D$ [illegible] (2.4) [illegible] $S + M^{-1}R^{-\top}DR^{-1}$ [illegible] (4.2) [illegible] $n$ [illegible]

Let $\widetilde{M} = (R^\top M R)^{-1}$. Then

$$S + M^{-1}R^{-\top}DR^{-1} = R(\Lambda + \widetilde{M}D)R^{-1}. \tag{4.4}$$

We need to find appropriate choices of $\widetilde{M}$ and $D$ such that $\Lambda + \widetilde{M}D$ has the $n$ given eigenvalues. We consider

$$\widetilde{M} = \operatorname{diag}\left(\begin{bmatrix} x_1 & z_1 \\ z_1 & y_1 \end{bmatrix}, \dots, \begin{bmatrix} x_s & z_s \\ z_s & y_s \end{bmatrix}\right) \tag{4.5}$$

for $n = 2s$ and

$$\widetilde{M} = \operatorname{diag}\left(\begin{bmatrix} x_1 & z_1 \\ z_1 & y_1 \end{bmatrix}, \dots, \begin{bmatrix} x_s & z_s \\ z_s & y_s \end{bmatrix}; \; x_{2s+1}\right) \tag{4.6}$$

for $n = 2s+1$. Consider the appropriate blocks in $\Lambda$, $\widetilde{M}$, and $D$ while ignoring indices; we only need to assign arbitrary (real or complex conjugate) eigenvalues to matrices of the forms

$$A_1 = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} + \begin{bmatrix} x & z \\ z & y \end{bmatrix}\begin{bmatrix} \xi & \eta \\ \eta & -\xi \end{bmatrix} \tag{4.7}$$

and

$$A_2 = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} + \begin{bmatrix} x & z \\ z & y \end{bmatrix}\begin{bmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{bmatrix}. \tag{4.8}$$

This can be achieved via appropriate choices of $x$, $y$, $z$, $\xi$, $\eta$, $\xi_1$, and $\xi_2$. We first require $x > 0$ and $xy - z^2 > 0$ so that $\widetilde{M} > 0$ as in (4.5) and (4.6).

(i) Assume that $A_1$ is of the form found in (4.7). In this case, we first choose $z = 0$. Then we prove that for any given real numbers $\tilde{\mu}$ and $\tilde{\nu}$, there are two positive numbers $x, y$ and two real number $\xi, \eta$ such that

$$\operatorname{tr} A_1 = \tilde{\mu}, \qquad \det A_1 = \tilde{\nu}.$$

This implies

$$\xi(x - y) = \tilde{\mu} - 2\alpha \equiv \mu, \tag{4.9}$$

$$(\alpha\xi + \beta\eta)(x - y) - (\xi^2 + \eta^2)xy = \tilde{\nu} - \alpha^2 - \beta^2 \equiv \nu. \tag{4.10}$$

If $\mu = 0$, we choose $\xi = 0$; then (4.9) holds. By taking $\eta = \frac{1}{\beta}$, (4.10) becomes

$$x(\beta^2 - y) = \beta^2(\nu + y). \tag{4.11}$$

Here we use the assumption that $\beta$ is nonzero. When $\nu = -\beta^2$, we take $y = \beta^2 > 0$; then (4.11) holds automatically. When $\nu > -\beta^2$, we take $y$ satisfying $\max\{-\nu, 0\} < y < \beta^2$; when $\nu < -\beta^2$, we take $y$ satisfying $\beta^2 < y < -\nu$. In both cases, we choose

$$x = \beta^2 \frac{\nu + y}{\beta^2 - y} > 0.$$

If $\mu \neq 0$, from (4.9) it follows that

$$x - y = \frac{\mu}{\xi}. \tag{4.12}$$

Substituting (4.12) into (4.10) leads to

$$xy = \frac{1}{\xi^2+\eta^2}\left(\mu\alpha - \nu + \frac{\eta}{\xi}\mu\beta\right) \equiv c. \tag{4.13}$$

Solving $x$ in (4.12) and substituting into (4.13), we get

$$y^2 + \frac{\mu}{\xi}y - c = 0. \tag{4.14}$$

It is easy to take $\xi$ and $\eta$ so that $c$, as defined in (4.13), is positive. Thus, from (4.14), we can take

$$y = \frac{1}{2}\left(-\frac{\mu}{\xi} + \sqrt{\left(\frac{\mu}{\xi}\right)^2 + 4c}\right) > 0, \tag{4.15}$$

and then, by (4.12), we have

$$x = \frac{\mu}{\xi} + y = \frac{1}{2}\left(\frac{\mu}{\xi} + \sqrt{\left(\frac{\mu}{\xi}\right)^2 + 4c}\right) > 0. \tag{4.16}$$

(ii) Assume that $A_2$ is of the form found in (4.8). We shall prove, for any given real numbers $\tilde{\mu}$ and $\tilde{\nu}$, that there exist positive numbers $x$, $y$ and real numbers $z$, $\xi_1$, $\xi_2$ which satisfy $xy - z^2 > 0$ and

$$\mathrm{tr} A_2 = \tilde{\mu}, \qquad \det A_2 = \tilde{\nu}.$$

This implies

$$\xi_1 x + \xi_2 y = \tilde{\mu} - \lambda_1 - \lambda_2 \equiv \mu, \tag{4.17}$$

$$\xi_1\lambda_2 x + \lambda_1\xi_2 y + \xi_1\xi_2(xy - z^2) = \tilde{\nu} - \lambda_1\lambda_2 \equiv \nu. \tag{4.18}$$

From (4.17), we have $\xi_1 = (\mu - \xi_2 y)/x$. Substituting it into (4.18), we get

$$\lambda_2(\mu - \xi_2 y) + \lambda_1\xi_2 y + \frac{1}{x}\xi_2(xy - z^2)(\mu - \xi_2 y) = \nu.$$

This implies

$$-(xy - z^2)\xi_2^2 + \left[(\lambda_1 - \lambda_2) + \mu\frac{xy - z^2}{xy}\right]\xi_2 + (\lambda_2\mu - \nu) = 0. \tag{4.19}$$

It remains to show that there are real numbers $x$, $y$, $z$ with $x > 0$ and $xy - z^2 > 0$ such that the quadratic equation (4.19) has real roots. This requires the discriminant of (4.19) to be positive, i.e.,

$$\left[(\lambda_1 - \lambda_2) + \mu\frac{xy - z^2}{xy}\right]^2 + 4(xy - z^2)(\lambda_2\mu - \nu) > 0. \tag{4.20}$$

To satisfy (4.20), we can first take $xy$ sufficiently large and then take $z$ such that $xy - z^2$ is a sufficiently small positive number. ∎

With $k = n$, we have constructed a particular solution to the PD-IQEP with $n$ additional eigenvalues.

ALGORITHM 4.1. SOLVING A PD-IQEP WITH $k = n$ AND $n$ ADDITIONAL EIGENVALUES.

[illegible] 1. Choose $M > 0$ and $D$ by (i) and (ii) as in Theorem 4.2 so that $S + M^{-1}R^{-\top}DR^{-1}$ has $n$ additionally given eigenvalues.

[illegible] 2. Compute $C$ and $K$ by (4.1a) and (4.1b), respectively.

[illegible] 3. Compute $M = QMQ^\top$, $C = QCQ^\top$, $K = QKQ^\top$, where $Q$ is given by (2.1).

**4.2. Particular solutions with additional eigenpairs.** In this subsection, we are interested in solving the PD-IQEP with $k = n$ and $r$ ($r \le \sqrt{n}$) additionally given eigenpairs. Note that with these $r$ ($r \le \sqrt{n}$) additional eigenpairs, this particular solution of the PD-IQEP is generically solvable (see Remark 4.1 later). Conversely, if $r > \sqrt{n}$, the PD-IQEP, in general, has no solution.

Suppose we are additionally given $r$ ($r \le \sqrt{n}$) eigenpairs $(\mu_j, \mathbf{y}_j) \in \mathbb{C} \times \mathbb{C}^n$, where

$$\mu_j = \bar{\mu}_{j+1} \in \mathbb{C}\backslash\mathbb{R}, \quad \mathbf{y}_j = \bar{\mathbf{y}}_{j+1} \in \mathbb{C}^n\backslash\mathbb{R}^n, \quad j = 1, 3, \ldots, 2s-1, \tag{4.21a}$$

$$\mu_j \in \mathbb{R}, \quad \mathbf{y}_j \in \mathbb{R}^n, \quad j = 2s+1, \ldots, r. \tag{4.21b}$$

Furthermore, we assume that $\mu_j \notin \sigma(\Lambda)$, $j = 1, \ldots, r$, and $[\mathbf{y}_1, \ldots, \mathbf{y}_r]$ is of full column rank. Then, for $r \le \sqrt{n}$, a quadratic pencil as in (4.2) always exists and satisfies $Q(\mu_j)\mathbf{y}_j = 0$ for $j = 1, \ldots, r$. To this end, we first prove a lemma.

LEMMA 4.1. [illegible] $\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r \in \mathbb{R}^n$ [illegible] $\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_r \in \mathbb{R}^n$ [illegible] $M$ [illegible]

$$M[\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r] = [\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_r] \tag{4.22}$$

[illegible]

$$[\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r]^\top [\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_r] \equiv Z^\top X > 0. \tag{4.23}$$

[illegible]. (Necessity.) If (4.22) holds, then

$$[\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r]^\top [\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_r] = [\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r]^\top M [\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r] > 0$$

because $M > 0$ and $Z = [\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r]$ is of full column rank.

(Sufficiency.) Let $V$ be an orthogonal matrix such that

$$V[\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_r] = \begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{4.24}$$

where $R \in \mathbb{R}^{r \times r}$ is nonsingular and upper triangular. Let

$$V[\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_r] = \left[\begin{array}{c} T_1 \\ \hline T_2 \end{array}\right], \quad T_1 \in \mathbb{R}^{r \times r}. \tag{4.25}$$

It is sufficient to find an $n \times n$ $\widetilde{M} > 0$ such that

$$\widetilde{M} \begin{bmatrix} R \\ 0 \end{bmatrix} = \left[\begin{array}{c} T_1 \\ \hline T_2 \end{array}\right]. \tag{4.26}$$

Then $M = V^\top \widetilde{M} V > 0$ satisfies (4.22). Partition $\widetilde{M}$ into

$$\widetilde{M} = \begin{bmatrix} \widetilde{M}_1 & \widetilde{M}_2 \\ \widetilde{M}_2^\top & \widetilde{M}_3 \end{bmatrix}.$$

By (4.26), we have $\widetilde{M}_1 = T_1 R^{-1}$. From (4.23)–(4.25), it follows that

$$\begin{aligned} R^\top \widetilde{M}_1 R &= R^\top T_1 R^{-1} R = R^\top T_1 \\ &= [R^\top, 0] \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = [R^\top, 0] V V^\top \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \\ &= Z^\top X > 0. \end{aligned}$$

Thus, $\widetilde{M}_1 > 0$. Taking $\widetilde{M}_2^\top = T_2 R^{-1}$ and $\widetilde{M}_3 = I + \widetilde{M}_2^\top \widetilde{M}_1^{-1} \widetilde{M}_2$, we have $\widetilde{M} > 0$. ☐

From (4.2), $Q(\mu_j)\mathbf{y}_j = 0$ is equivalent to

$$\left(\mu_j I - S^\top - R^{-T} D R^{-1} M^{-1}\right) M \mathbf{z}_j = 0, \tag{4.27}$$

where $\mathbf{z}_j = (\mu_j I - S)\mathbf{y}_j$, $j = 1, \dots, r$. By letting $\mathbf{v}_j = R^{-1}\mathbf{z}_j \equiv [v_{j,1}, \dots, v_{j,n}]^\top$ and $\widetilde{M} = R^\top M R$, (4.27) is equivalent to

$$\widetilde{M}\mathbf{v}_j = (\mu_j I - \Lambda^\top)^{-1} D \mathbf{v}_j, \quad j = 1, \dots, r. \tag{4.28}$$

We now define an $n$-vector, corresponding to $D$ in (2.4), by

$$\mathbf{d} = [\xi_1,\ \eta_1; \dots ; \xi_\ell,\ \eta_\ell;\ \xi_{2\ell+1}, \dots, \xi_n]^\top. \tag{4.29}$$

Write

$$D\mathbf{v}_j = V_j \mathbf{d}, \quad j = 1, \dots, r, \tag{4.30}$$

where

$$V_j = \operatorname{diag}\left( \begin{bmatrix} v_{j,1} & v_{j,2} \\ -v_{j,2} & v_{j,1} \end{bmatrix}, \dots, \begin{bmatrix} v_{j,2\ell-1} & v_{j,2\ell} \\ -v_{j,2\ell} & v_{j,2\ell-1} \end{bmatrix}; v_{j,2\ell+1}, \dots, v_{j,n} \right).$$

Then (4.28) implies

$$\widetilde{M}\mathbf{v}_j = (\mu_j I - \Lambda^\top)^{-1} V_j \mathbf{d}, \qquad j = 1, \dots, r. \tag{4.31}$$

Let

$$\boldsymbol{\zeta}_j = \operatorname{Re}(\mathbf{v}_j), \qquad \boldsymbol{\zeta}_{j+1} = \operatorname{Im}(\mathbf{v}_j), \qquad j = 1, 3, \dots, 2s-1, \tag{4.32a}$$

$$G_j = \operatorname{Re}\left((\mu_j I - \Lambda^\top)^{-1} V_j\right), \qquad G_{j+1} = \operatorname{Im}\left((\mu_j I - \Lambda^\top)^{-1} V_j\right), \quad j = 1, 3, \dots, 2s-1, \tag{4.32b}$$

$$\boldsymbol{\zeta}_j = \mathbf{v}_j, \qquad G_j = (\mu_j I - \Lambda^\top)^{-1} V_j, \qquad j = 2s+1, \dots, r. \tag{4.32c}$$

From (4.31)–(4.32) and Lemma 4.1, we proved the following theorem.

THEOREM 4.3. [illegible] $(\mu_j, \mathbf{y}_j) \in \mathbb{C} \times \mathbb{C}^n$ $(j = 1, \dots, r)$ [illegible] (4.21) [illegible] $M$ [illegible] $D$ [illegible] (2.4) [illegible] $Q(\mu_j)\mathbf{y}_j = 0$ $(j = 1, \dots, r)$ [illegible]

$$W \equiv [\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_r]^\top [G_1 \mathbf{d}, \dots, G_r \mathbf{d}] > 0, \tag{4.33}$$

[illegible] $\boldsymbol{\zeta}_j$ [illegible] $G_j$ $(j = 1, \dots, r)$ [illegible] (4.32) [illegible] $\mathbf{d}$ [illegible] (4.29) [illegible] 4.1. Suppose $r \leq \sqrt{n}$. If we set $W$ in (4.33) to be any symmetric positive definite matrix, then (4.33) can form a squared or underdetermined real linear system in $\mathbf{d} \in \mathbb{R}^n$:

$$\boldsymbol{\zeta}_i^\top G_j \mathbf{d} = w_{ij}, \quad i, j = 1, \ldots, r \leq \sqrt{n}. \tag{4.34}$$

Thus, the vector $\mathbf{d} \in \mathbb{R}^n$, and therefore $D$ in (2.4), are generically solvable.

[illegible] 4.2. The solution of the monic IQEP in [4] is a special case of Theorem 4.3 with $r = 1$.

With $k = n$, we have constructed a particular solution to the PD-IQEP with $r$ ($\leq \sqrt{n}$) additional eigenpairs.

ALGORITHM 4.2. SOLVING A PD-IQEP WITH $k = n$ AND $r$ ($r \leq \sqrt{n}$) ADDITIONAL EIGENPAIRS.

[illegible] 1. Compute $\boldsymbol{\zeta}_j$ and $G_j$ in (4.32) for $j = 1, \ldots, r$.

[illegible] 2. Choose a $W \equiv [w_{ij}]_{r \times r} > 0$; if the linear equation (4.34) for $\mathbf{d}$ is solvable, then go to Step 3; else repeat Step 2.

[illegible] 3. Compute $M > 0$ as in Lemma 4.1 by setting $\boldsymbol{\delta}_j = G_j \mathbf{d}$, $j = 1, \ldots, r$.

[illegible] 4. Compute $C$ and $K$ by (4.1a) and (4.1b), respectively.

[illegible] 5. Compute $M = QMQ^\top$, $C = QCQ^\top$, $K = QKQ^\top$, where $Q$ is given by (2.1).

**5. Particular solutions with additional eigeninformation when $\boldsymbol{k < n}$.** When $k < n$, for a given matrix pair $(\Lambda, X) \in \mathbb{R}^{k \times k} \times \mathbb{R}^{n \times k}$ as in (1.2) and (1.3) and under assumptions (H1)–(H2), Theorem 2.1 states that the general solution to the PD-IQEP is

$$M = Q \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} Q^\top, \; C = Q \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} Q^\top, \; K = Q \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} Q^\top, \tag{5.1}$$

where $M$, $C_{22}$, $C_{21} = C_{12}^\top$ and $K_{22}$ can be arbitrarily chosen, while $C_{11}$, $K_{11}$, and $K_{21} = K_{12}^\top$ are given by (2.3). Furthermore, we have the following.

THEOREM 5.1. [illegible] $k < n$ [illegible]

$$\begin{aligned} &Q(\lambda) = \lambda^2 M + \lambda C + K \\ &= \begin{bmatrix} \left(\lambda I - S^\top - R^{-T} D R^{-1} M_{11}^{-1}\right) M_{11} & (\lambda I - S^\top)\left(M_{21}(\lambda I + S) + C_{21}\right)^\top \\ \left(M_{21}(\lambda I + S) + C_{21}\right) & \lambda^2 M_{22} + \lambda C_{22} + K_{22} \end{bmatrix} \begin{bmatrix} \lambda I - S & 0 \\ 0 & I \end{bmatrix}, \end{aligned} \tag{5.2}$$

[illegible] $Q(\lambda)$ [illegible] $\Lambda$

[illegible]. From (5.1) and (2.3), we have

$$\begin{aligned} \lambda^2 M_{11} + \lambda C_{11} + K_{11} &= \left(\lambda I - S^\top - R^{-T} D R^{-1} M_{11}^{-1}\right) M_{11} (\lambda I - S), \\ \lambda^2 M_{21} + \lambda C_{21} + K_{21} &= M_{21}(\lambda^2 I - S^2) + C_{21}(\lambda I - S) \\ &= [M_{21}(\lambda I + S) + C_{21}] (\lambda I - S), \end{aligned}$$

which imply (5.2). ☐

In the following subsections, we construct two particular solutions to the PD-IQEP when $k < n$, with one additional complex eigenpair (with $k = n - 1$) or positive definite property of $M$ and $K$. These solutions are equivalent to those developed in [4].

**5.1. Particular solutions with an additional complex eigenpair.** In this subsection, we shall construct the particular solution of the PD-IQEP with $k = n-1$ and an additional complex eigenpairs. We shall show that this particular solution is equivalent to the unique solution of the monic IQEP in [4] with $M = I_n$.

For a given pair $(\Lambda, X) \in \mathbb{R}^{(n-1)\times(n-1)} \times \mathbb{R}^{n\times(n-1)}$ as in (1.2) and (1.3), under assumptions (H1)–(H2), Theorem 5.1 shows that the general solution of the PD-IQEP has the decomposition (5.2), where $S = R\Lambda R^{-1}$ and $Q^\top X = [R^\top, 0]^\top$. For an additionally given complex eigenpair $(\mu, \mathbf{z})$ with $\mu \notin \sigma(\Lambda)$, we want to solve the PD-IQEP by (2.3) where $M = I_n$ and $Q(\mu)\mathbf{z} = Q(\bar{\mu})\bar{\mathbf{z}} = 0$.

Write $\mathbf{z} = \left[\begin{smallmatrix} \mathbf{r}_{12} \\ r_{22} \end{smallmatrix}\right]$, where $\mathbf{r}_{12} \in \mathbb{C}^{n-1}$ and $0 \neq r_{22} \in \mathbb{C}$, and denote $\widehat{R} = \left[\begin{smallmatrix} R & \mathbf{r}_{12} \\ 0 & r_{22} \end{smallmatrix}\right]$ and $\widehat{\Lambda} = \left[\begin{smallmatrix} \Lambda & 0 \\ 0 & \mu \end{smallmatrix}\right]$. Then we have

$$\widehat{S} \equiv \widehat{R}\widehat{\Lambda}\widehat{R}^{-1} = \begin{bmatrix} R\Lambda R^{-1} & -r_{22}^{-1} R\Lambda R^{-1}\mathbf{r}_{12} + \mu r_{22}^{-1}\mathbf{r}_{12} \\ 0 & \mu \end{bmatrix}. \tag{5.3}$$

In [4], the general solution of the Hermitian matrix $\widehat{C}$ in $Q(\lambda) \equiv \lambda^2 \widehat{M} + \lambda\widehat{C} + \widehat{K}$ is given by

$$\begin{aligned} \widehat{C} &\equiv \begin{bmatrix} \widehat{C}_{11} & \hat{\mathbf{c}}_{12} \\ \hat{\mathbf{c}}_{12}^H & \hat{c}_{22} \end{bmatrix} \\ &= -\left(\widehat{S} + \widehat{S}^H\right) - \left[\begin{array}{c} R^{-\top} \\ \hline -\bar{r}_{22}^{-1}\mathbf{r}_{12}^H R^{-\top} \end{array}\right] D \left[\begin{array}{c|c} R^{-1} & -r_{22}^{-1}R^{-1}\mathbf{r}_{12} \end{array}\right], \end{aligned} \tag{5.4}$$

where $D$ is of the form in (2.4), $\widehat{C}_{11} \in \mathbb{R}^{(n-1)\times(n-1)}$, $\hat{\mathbf{c}}_{12} \in \mathbb{C}^{n-1}$, and $\hat{c}_{22} \in \mathbb{C}$. Expanding $\widehat{C}_{11}$, $\hat{\mathbf{c}}_{12}$, and $\hat{c}_{22}$ in (5.4), we get

$$\widehat{C}_{11} = -\left(S + S^\top\right) - R^{-\top}DR^{-1}, \tag{5.5}$$

$$\hat{\mathbf{c}}_{12} = -r_{22}^{-1}\left(\mu I - S\right)\mathbf{r}_{12} + r_{22}^{-1}R^{-\top}DR^{-1}\mathbf{r}_{12}, \tag{5.6}$$

$$\hat{c}_{22} = -\mu - \bar{\mu} - |r_{22}|^{-2}\mathbf{r}_{12}^H R^{-\top}DR^{-1}\mathbf{r}_{12}. \tag{5.7}$$

On the other hand, by setting $M_{11} = I_{n-1}$ and $M_{21} = 0$ in (5.2) and relaxing $C$ to be Hermitian, we have

$$\begin{aligned} Q(\lambda) &\equiv \lambda^2 I + \lambda C + K \\ &= \begin{bmatrix} \lambda I - S^\top - R^{-\top}DR^{-1} & (\lambda I - S^\top)\mathbf{c}_{12} \\ \mathbf{c}_{12}^H & \lambda^2 + \lambda c_{22} + k_{22} \end{bmatrix} \begin{bmatrix} \lambda I - S & 0 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{5.8}$$

Here we partition $C$ as

$$C = \begin{bmatrix} C_{11} & \mathbf{c}_{12} \\ \mathbf{c}_{12}^H & c_{22} \end{bmatrix}, \tag{5.9}$$

where $C_{11} \in \mathbb{R}^{(n-1)\times(n-1)}$, $\mathbf{c}_{12} \in \mathbb{C}^{n-1}$, $c_{22} \in \mathbb{C}$.

We want to show C $= \widehat{C}$ when $Q(\mu)\mathbf{z} = Q(\bar{\mu})\bar{\mathbf{z}} = 0$. From [4], $C$ in (5.9) satisfies $C = \bar{C} = C^H = C^\top$, and therefore $K = K^\top$.

It is easily seen that

$$C_{11} = -(S + S^\top) - R^{-\top}DR^{-1} = \widehat{C}_{11}. \tag{5.10}$$

Substituting $\lambda = \mu$ in (5.8), we compute $Q(\mu)\mathbf{z} \equiv Q(\mu)\left[\begin{smallmatrix}\mathbf{r}_{12}\\ r_{22}\end{smallmatrix}\right] = 0$ and obtain

$$(\mu I - S^\top - R^{-\top} D R^{-1})(\mu I - S)\mathbf{r}_{12} + r_{22}(\mu I - S^\top)\mathbf{c}_{12} = 0, \tag{5.11}$$

$$\mathbf{c}_{12}^H(\mu I - S)\mathbf{r}_{12} + r_{22}(\mu^2 + \mu c_{22} + k_{22}) = 0. \tag{5.12}$$

Then from $D\Lambda = \Lambda^\top D$, we have

$$\begin{aligned}\mathbf{c}_{12} &= -r_{22}^{-1}(\mu I - S^\top)^{-1}(\mu I - S^\top - R^{-\top} D R^{-1})(\mu I - S)\mathbf{r}_{12}\\ &= -r_{22}^{-1}(\mu I - S)\mathbf{r}_{12} + r_{22}^{-1} R^{-\top}(\mu I - \Lambda)^{-\top} D(\mu I - \Lambda)R^{-1}\mathbf{r}_{12}\\ &= -r_{22}^{-1}(\mu I - S - R^{-\top} D R^{-1})\mathbf{r}_{12} = \hat{\mathbf{c}}_{12}.\end{aligned} \tag{5.13}$$

Similarly, from $Q(\bar\mu)\left[\begin{smallmatrix}\bar{\mathbf{r}}_{12}\\ \bar r_{22}\end{smallmatrix}\right] = 0$, we have

$$(\bar\mu I - S^\top - R^{-\top} D R^{-1})(\bar\mu I - S)\bar{\mathbf{r}}_{12} + \bar r_{22}(\bar\mu I - S^\top)\mathbf{c}_{12} = 0, \tag{5.14}$$

$$\mathbf{c}_{12}^H(\bar\mu I - S)\bar{\mathbf{r}}_{12} + \bar r_{22}(\bar\mu^2 + \bar\mu c_{22} + k_{22}) = 0. \tag{5.15}$$

Eliminating $k_{22}$ in (5.12) using the difference of (5.12) and (5.15), we get

$$(\mu - \bar\mu)c_{22} + \mathbf{c}_{12}^H[r_{22}^{-1}(\mu I - S)\mathbf{r}_{12} - \bar r_{22}^{-1}(\bar\mu I - S)\bar{\mathbf{r}}_{12}] + \mu^2 - \bar\mu^2 = 0. \tag{5.16}$$

From (5.11) and (5.13)–(5.14), it follows that

$$\begin{aligned}(\mu - \bar\mu)c_{22} &= \mathbf{c}_{12}^H[\bar r_{22}^{-1}(\bar\mu I - S)\bar{\mathbf{r}}_{12} - r_{22}^{-1}(\mu I - S)\mathbf{r}_{12}] + \bar\mu^2 - \mu^2\\ &= -\bar r_{22}^{-1}\mathbf{r}_{12}^H(\bar\mu I - S^\top - R^{-\top} D R^{-1})[\bar r_{22}^{-1}(\bar\mu I - S)\bar{\mathbf{r}}_{12} - r_{22}^{-1}(\mu I - S)\mathbf{r}_{12}]\\ &\quad + \bar\mu^2 - \mu^2 \qquad\qquad \text{(from (4.14))}\\ &= \bar r_{22}^{-1}\mathbf{r}_{12}^H[\mu I - S^\top - R^{-\top} D R^{-1} + (\bar\mu - \mu)I]r_{22}^{-1}(\mu I - S)\mathbf{r}_{12}\\ &\quad - \bar r_{22}^{-1}\mathbf{r}_{12}^H[-(\bar\mu I - S^\top)\mathbf{c}_{12}] + \bar\mu^2 - \mu^2 \qquad\qquad \text{(from (4.11))}\\ &= -\bar r_{22}^{-1}\mathbf{r}_{12}^H[-(\bar\mu I - S^\top)\mathbf{c}_{12} + (\mu - S^\top)\mathbf{c}_{12} - (\bar\mu - \mu)r_{22}^{-1}(\mu I - S)\mathbf{r}_{12}]\\ &\quad + \bar\mu^2 - \mu^2\\ &= -\bar r_{22}^{-1}\mathbf{r}_{12}^H[(\mu - \bar\mu)(\mathbf{c}_{12} + r_{22}^{-1}(\mu I - S)\mathbf{r}_{12}] + \bar\mu^2 - \mu^2 \qquad \text{(from (4.13))}\\ &= -\bar r_{22}^{-1}\mathbf{r}_{12}^H[r_{22}^{-1}(\mu - \bar\mu)R^{-\top} D R^{-1}\mathbf{r}_{12}] + \bar\mu^2 - \mu^2\\ &= (\mu - \bar\mu)(-|r_{22}|^{-2}\mathbf{r}_{12}^H R^{-\top} D R^{-1}\mathbf{r}_{12} - \mu - \bar\mu).\end{aligned}$$

Hence $c_{22} = \hat c_{22}$. Combining with (5.10) and (5.13), we have shown that $C = \widehat{C}$.

**5.2. Particular solutions with special structures.** In this subsection, we shall construct a particular solution of the PD-IQEP with $k < n$, $M > 0$, and $K > 0$. Under suitable condition this solution is equivalent to the solution of the standard IQEP developed in [4].

We now take $D = 0$; then the decomposition (5.2) becomes

(5.17)

$$Q(\lambda) = \begin{bmatrix}\lambda I - S & 0\\ 0 & I\end{bmatrix}^\top \begin{bmatrix} M_{11} & (M_{21}(\lambda I + S) + C_{21})^\top\\ M_{21}(\lambda I + S) + C_{21} & \lambda^2 M_{22} + \lambda C_{22} + K_{22}\end{bmatrix}\begin{bmatrix}\lambda I - S & 0\\ 0 & I\end{bmatrix}.$$

Thus, we have

$$\det(Q(\lambda)) = [\det(\lambda I - S)]^2 \det(Q_2(\lambda)), \tag{5.18}$$

where

$$\begin{aligned} Q_2(\lambda) &= \lambda^2 M_{22} + \lambda C_{22} + K_{22} - [M_{21}(\lambda I + S) + C_{21}]\, M_{11}^{-1}\, [M_{21}(\lambda I + S) + C_{21}]^\top \\ &\equiv \lambda^2 \widetilde{M}_{22} + \lambda \widetilde{C}_{22} + \widetilde{K}_{22}, \end{aligned} \tag{5.19}$$

with

$$\widetilde{M}_{22} = M_{22} - M_{21} M_{11}^{-1} M_{12}^\top, \tag{5.20a}$$

$$\widetilde{C}_{22} = C_{22} - M_{21} M_{11}^{-1} (M_{21}S + C_{21})^\top - (M_{21}S + C_{21}) M_{11}^{-1} M_{21}^\top, \tag{5.20b}$$

$$\widetilde{K}_{22} = K_{22} - (M_{21}S + C_{21}) M_{11}^{-1} (M_{21}S + C_{21})^\top. \tag{5.20c}$$

On the other hand, with $m = 2n - k > n$, choose an arbitrary matrix $U \in \mathbb{R}^{m\times n}$ which is of full column rank. Partition $U = [U_1, U_2]$ with $U_1 \in \mathbb{R}^{m\times k}$. Thus, if we take

$$M = U^\top U > 0, \tag{5.21}$$

we obtain

$$M_{11} = U_1^\top U_1,\ M_{21} = U_2^\top U_1,\ M_{22} = U_2^\top U_2. \tag{5.22}$$

Substituting (5.22) into (2.3), we have

$$C_{11} = -(U_1^\top U_1 S + S^\top U_1^\top U_1), \tag{5.23a}$$

$$K_{11} = S^\top U_1^\top U_1 S, \tag{5.23b}$$

$$K_{21} = -(U_2^\top U_1 S + C_{21}) S. \tag{5.23c}$$

Let $V_1 = -U_1 S$ and $V_2$ be an arbitrary $m \times (n - k)$ matrix. Taking $C_{21} = U_2^\top V_1 + V_2^\top U_1$ and substituting it into (5.23) lead to

$$C_{11} = U_1^\top V_1 + V_1^\top U_1, \quad K_{11} = V_1^\top V_1, \tag{5.24a}$$

$$K_{21} = (U_2^\top V_1 - U_2^\top V_1 - V_2^\top U_1) S = V_2^\top V_1. \tag{5.24b}$$

If we write $V = [V_1, V_2]$ and take

$$C_{22} = U_2^\top V_2 + V_2^\top U_2, \quad K_{22} = V_2^\top V_2, \tag{5.25}$$

then from (5.24) we obtain

$$C = U^\top V + V^\top U, \quad K = V^\top V. \tag{5.26}$$

Assume without loss of generality that $X = [R^\top, 0]^\top$, and with $V_1 = -U_1 S$, we then have

$$[R^\top, \Lambda^\top R^\top] \begin{bmatrix} V_1^\top \\ U_1^\top \end{bmatrix} = [X^\top, \Lambda^\top X^\top] \begin{bmatrix} V^\top \\ U^\top \end{bmatrix} = 0. \tag{5.27}$$

If $V_2$ above is chosen so that $\left[\begin{smallmatrix} V^\top \\ U^\top \end{smallmatrix}\right]$ is of full column rank, then the quadratic pencil defined by (5.21) and (5.26) is precisely the solution of the standard IQEP in [4].

Furthermore, from (5.22)–(5.25), the matrices in (5.20) become

$$\widetilde{M}_{22} = U_2^\top B U_2, \tag{5.28a}$$

$$\widetilde{C}_{22} = U_2^\top B V_2 + V_2^\top B U_2, \tag{5.28b}$$

$$\widetilde{K}_{22} = V_2^\top B V_2, \tag{5.28c}$$

where $B = I - U_1(U_1^\top U_1)^{-1}U_1^\top$. Therefore, $Q_2(\lambda)$ in (5.18) becomes

$$\begin{aligned} Q_2(\lambda) &= \lambda^2 \widetilde{M}_{22} + \lambda \widetilde{C}_{22} + \widetilde{K}_{22} \\ &= (\lambda U_2 + V_2)^\top B (\lambda U_2 + V_2). \end{aligned} \tag{5.29}$$

Note that here $B$ is an orthogonal projector, i.e., $B^2 = B = B^\top$.

We now consider the QR-decomposition

$$U = [U_1, U_2] = P \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \\ 0 & 0 \end{bmatrix}, \tag{5.30}$$

where $P$ is orthogonal, and $T_{11} \in \mathbb{R}^{k\times k}$ and $T_{22} \in \mathbb{R}^{(n-k)\times(n-k)}$ are both nonsingular and upper triangular. Then

$$V_1 = -U_1 S = -P \begin{bmatrix} T_{11}S \\ 0 \\ 0 \end{bmatrix}. \tag{5.31}$$

Let

$$P^\top V_2 = \begin{array}{c c} n-k & \\ \left[\begin{matrix} T_{13} \\ T_{23} \\ T_{33} \end{matrix}\right] & \begin{matrix} k \\ n-k \\ m-n = n-k \end{matrix} \end{array}. \tag{5.32}$$

As $\left[\begin{smallmatrix} V^\top \\ U^\top \end{smallmatrix}\right]$ is of full column rank, it follows that $T_{33}$ is nonsingular. Therefore, we have

$$B = I - U_1(U_1^\top U_1)^{-1}U_1^\top = P \begin{bmatrix} 0 & 0 \\ 0 & I_{n-k} \end{bmatrix} P^\top$$

and

$$\lambda U_2 + V_2 = P \begin{bmatrix} \lambda T_{12} + T_{13} \\ \lambda T_{22} + T_{23} \\ T_{33} \end{bmatrix},$$

implying

$$Q_2(\lambda) = (\lambda T_{22} + T_{23})^\top (\lambda T_{22} + T_{23}) + T_{33}^\top T_{33}. \tag{5.33}$$

Because $T_{33}$ is nonsingular, $\det Q_2(\lambda) > 0$ for any real number $\lambda$. Thus, $Q_2(\lambda)$ in (5.18) has only complex conjugate eigenvalues with nonzero imaginary part. Furthermore, if $\left[\begin{smallmatrix} V^\top \\ U^\top \end{smallmatrix}\right]$ is chosen to be orthogonal, then we have $T_{23} = 0$. From (5.33), we have $Q_2(\lambda) = (\lambda^2 + 1)I$ with eigenvalues $\lambda = \pm\boldsymbol{\iota}$. This also coincides with the result in [4].

Table 6.1
*Absolute errors of computed eigenvalues.*

| Eigenvalues | Absolute error $\lvert\lambda_i - \widehat{\lambda}_i\rvert$ |
|---|---|
| $\lambda_1 = \bar{\lambda}_2$ | 1.2766e-011 |
| $\lambda_3$ | 4.2322e-013 |
| $\lambda_4$ | 2.7001e-013 |
| $\lambda_5$ | 9.7161e-013 |
| $\lambda_6$ | 7.6428e-012 |
| $\lambda_7 = \bar{\lambda}_8$ | 9.9151e-012 |
| $\lambda_9 = \bar{\lambda}_{10}$ | 3.5711e-010 |
| $\lambda_{11}$ | 2.2284e-010 |
| $\lambda_{12}$ | 6.1142e-012 |

**6. Numerical examples.** The results presented in sections 4 and 5 offer a constructive way to solve the PD-IQEP with additional eigeninformation or special structures. In this section, we present two numerical examples to illustrate the particular solutions constructed in section 4. Numerical examples constructed by section 5 can be found in [4]. For presentation, we report all numbers in 5 significant digits only, though all calculations are carried out in full precision.

To generate test data, we first randomly generate the partially prescribed eigeninformation $(\Lambda, X) \in \mathbb{R}^{6\times 6} \times \mathbb{R}^{6\times 6}$ as in (1.2) and (1.3), with $\lambda_1 = 3.5121 + 8.2485\boldsymbol{\imath} = \bar{\lambda}_2$, $\lambda_3 = 1.7541$, $\lambda_4 = 1.2596$, $\lambda_5 = 0.42402$, $\lambda_6 = 6.4268$, and the corresponding eigenvectors

$$\mathbf{x}_1 = \bar{\mathbf{x}}_2 = \begin{bmatrix} 9.2963 + 1.0480\boldsymbol{\imath} \\ 2.3695 + 3.5650\boldsymbol{\imath} \\ 3.8789 + 6.5809\boldsymbol{\imath} \\ 2.8644 + 4.9742\boldsymbol{\imath} \\ 1.5007 + 1.1356\boldsymbol{\imath} \\ 1.9623 + 6.5805\boldsymbol{\imath} \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 8.3476 \\ 8.0904 \\ 5.5542 \\ 9.2809 \\ 4.0705 \\ 2.8111 \end{bmatrix},$$

$$\mathbf{x}_4 = \begin{bmatrix} 6.6044 \\ 9.3147 \\ 2.5443 \\ 2.1294 \\ 8.1057 \\ 2.7021 \end{bmatrix}, \quad \mathbf{x}_5 = \begin{bmatrix} 3.7708 \\ 5.8997 \\ 6.7357 \\ 6.6171 \\ 3.7162 \\ 8.6814 \end{bmatrix}, \quad \mathbf{x}_6 = \begin{bmatrix} 8.5856 \\ 8.3937 \\ 3.3068 \\ 8.1301 \\ 1.8702 \\ 2.7268 \end{bmatrix}.$$

It is easy to check that the matrix pair $(\Lambda, X) \in \mathbb{R}^{6\times 6} \times \mathbb{R}^{6\times 6}$ satisfies the assumptions (H1) and (H2).

[illegible] 1. Subsection 4.1 shows that the PD-IQEP may have 6 assigned eigenvalues, closed under complex conjugation. We generate randomly 6 eigenvalues: $\lambda_7 = 4.6983 + 4.0385\boldsymbol{\imath} = \bar{\lambda}_8$, $\lambda_9 = 5.6495 + 1.5612\boldsymbol{\imath} = \bar{\lambda}_{10}$, $\lambda_{11} = 8.9227$, $\lambda_{12} = 4.1238$. We compute a real symmetric quadratic pencil

$$\widehat{Q}(\lambda) = \lambda^2 \widehat{M} + \lambda \widehat{C} + \widehat{K}$$

by using the method described in subsection 4.1. The residual $\|\widehat{M}X\Lambda^2 + \widehat{C}X\Lambda + \widehat{K}X\|_2$ and the eigenvalues of $\widehat{M}$ are, respectively, 3.6953e-011 and $\{150.52,\ 5.9347,\ 0.12706, 1.8466\text{e-}2,\ 1.2481\text{e-}2,\ 5.5983\text{e-}4\}$. In Table 6.1, we display the absolute errors $|\lambda_i - \widehat{\lambda}_i|$ $(i = 1, \ldots, 12)$, where $\widehat{\lambda}_i$ are the computed eigenvalues of $\widehat{Q}(\lambda)$.

These numerical results show that the computed quadratic pencil is satisfactory.

2. Subsection 4.2 shows the PD-IQEP may have 2 ($< \sqrt{6}$) additional assigned eigenpairs, closed under complex conjugation. We randomly generate 2 eigenpairs: $\lambda_7 = 4.6983 + 4.0385\boldsymbol{\iota} = \bar{\lambda}_8$ and

$$\mathbf{x}_7 = \bar{\mathbf{x}}_8 = \begin{bmatrix} 8.6239 + 6.7913\boldsymbol{\iota} \\ 1.2742 + 1.5386\boldsymbol{\iota} \\ 6.0538 + 9.0277\boldsymbol{\iota} \\ 5.4385 + 8.5876\boldsymbol{\iota} \\ 4.6188 + 1.0069\boldsymbol{\iota} \\ 7.9774 + 3.1084\boldsymbol{\iota} \end{bmatrix}.$$

We compute a real symmetric quadratic pencil

$$\check{Q}(\lambda) = \lambda^2 \check{M} + \lambda \check{C} + \check{K}$$

by using the method described in subsection 4.2. We have the following numerical results:

$$\begin{aligned} &\|\check{M} X \Lambda^2 + \check{C} X \Lambda + \check{K} X\|_2 = 2.0916\text{e-}013, \\ &\|\check{Q}(\lambda_7)\mathbf{x}_7\|_2 = \|\check{Q}(\lambda_8)\mathbf{x}_8\|_2 = 1.6213\text{e-}013, \\ &\sigma(\check{M}) = \{1.0144,\ 1.0081,\ 1,\ 1,\ 3.4592\text{e-}3,\ 1.1868\text{e-}3\}. \end{aligned}$$

This shows that the matrix $\check{M}$ is symmetric positive definite and the residuals are small.

**7. Conclusions.** In this paper, we use techniques involving matrix decompositions to derive an expression of the general solution to the PD-IQEP, for a set of given $k$ eigenpairs ($k \leq n$), under assumptions (H1) and (H2). With appropriate choices of degrees of freedom, we can construct a quadratic pencil $Q(\lambda) = \lambda^2 M + \lambda C + K$ with $M > 0$ and $K \geq 0$. Furthermore, we can also find solutions which satisfy various additional eigeninformation, as shown in sections 4 and 5. The problem of how to utilize the degrees of freedom in general, or under other given sets of eigeninformation, is interesting and needs further investigation. In summary, we list some of the solved and unsolved PD-IQEPs, under various constraints, in Tables 7.1 and 7.2.

For another case of $k > n$, it is rather involved and the proof technique of Theorem 2.1 seems not to be used directly to find a general solution of PD-IQEP. To our knowledge, this case has never been discussed in the literature. It might be interesting research and needs further investigation.

TABLE 7.1

*PD-IQEP solved in section* 2.

| |
|---|
| Given $\Lambda$ and $X$ as in (1.2) and (1.3) under assumptions (H1) and (H2), equations (2.2)–(2.4) give rise to symmetric $M$, $C$, and $K$ with $M > 0$ such that (1.4) holds. |

**Acknowledgment.** We would like to give our best thanks to Professor Moody Chu (NCSU) for many valuable comments and helpful discussions on this paper.

TABLE 7.2

*List of solved and unsolved PD-IQEPs with additional conditions.*

| Number of PD-eigenpairs | Additional eigeninformation | Status |
|---|---|---|
| $k \leq n$ | $K \geq 0$ | Solved in section 3 |
| $k = n$ | $n$ additional eigenvalues | Solved in section 4.1 |
| $k = n$ | $r$ $(r \leq \sqrt{n})$ additional eigenpairs | Solved in section 4.2 |
| $k = n - 1$ | One additional complex eigenpair | Solved in [4] or section 5.1 |
| $k < n$ | $D = 0$ in (2.3) and $K > 0$ | Solved in [4] or section 5.2 |
| $k = n$ | $r$ $(r \leq \sqrt{n})$ additional eigenpairs and $K \geq 0$ | Unsolved |
| $k = n$ | $\frac{n}{2}$ additional specified eigenpairs | Unsolved |
| $k \leq n$ | $\min\{\mu\|M-M_a\|_F^2+\nu\|C-C_a\|_F^2+\|K-K_a\|_F^2 :$ $M > 0,\ K=K^\top,\ C=C^\top\}$,<br>where $M_a$, $C_a$, $K_a$ are given analytic models and $\mu$, $\nu$ are appropriate parameters. | Unsolved |

## REFERENCES

[1] J. Carvalho, B. N. Datta, W. W. Lin, and C. S. Wang, *Symmetric preserving eigenvalue embedding in finite element model updating of vibrating structures*, J. Sound Vibration, 290 (2006), pp. 839–864.

[2] E. K.-W. Chu and B. N. Datta, *Numerically robust pole assignment for second-order systems*, Internat. J. Control, 64 (1996), pp. 1113–1127.

[3] M. T. Chu and G. H. Golub, *Structured inverse eigenvalue problems*, Acta Numer., 11 (2002), pp. 1–71.

[4] M. T. Chu, Y.-C. Kuo, and W.-W. Lin, *On inverse quadratic eigenvalue problems with partially prescribed eigenstructure*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 995–1020.

[5] M. T. Chu, *Inverse eigenvalue problems*, SIAM Rev., 40 (1998), pp. 1–39.

[6] B. N. Datta, S. Elhay, Y. M. Ram, and D. R. Sarkissian, *Partial eigenstructure assignment for the quadratic pencil*, J. Sound Vibration, 230 (2000), pp. 101–110.

[7] B. N. Datta, *Finite element model updating, eigenstructure assignment and eigenvalue embedding techniques for vibrating systems, mechanical systems and signal processing*, Mech. Systems Signal Process., 16 (2002), pp. 83–96.

[8] M. I. Friswell, D. J. Inman, and D. F. Pilkey, *The direct updating of damping and stiffness matrices*, AIAA J., 36 (1998), pp. 491–493.

[9] M. I. Friswell and J. E. Mottershead, *Finite Element Model Updating in Structural Dynamics*, Kluwer Academic, Dordrecht, The Netherlands, 1995.

[10] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[11] Y. C. Kuo, W. W. Lin, and S. F. Xu, *A new model correcting method for quadratic eigenvalue problems using a symmetric eigenstructure assignment*, AIAA J., 43 (2005), pp. 2593–2598.

[12] P. Lancaster and U. Prells, *Inverse problems for damped vibrating systems*, J. Sound Vibration, 283 (2005), pp. 891–914.

[13] W. W. Lin and J. N. Wang, *Partial pole assignment for the quadratic pencil by output feedback control with feedback designs*, Numer. Linear Algebra Appl., 12 (2005), pp. 967–979.

[14] N. K. Nichols and J. Kautsky, *Robust eigenstructure assignment in quadratic matrix polynomials: Nonsingular case*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 77–102.

[15] Y. M. Ram and S. Elhay, *An inverse eigenvalue problem for the symmetric tridiagonal quadratic pencil with application to damped oscillatory systems*, SIAM J. Appl. Math., 56 (1996), pp. 232–244.

[16] D. D. Sivan and Y. M. Ram, *Physical modifications to vibratory systems with assigned eigendata*, ASME J. Appl. Mech., 66 (1999), pp. 427–432.

[17] L. Starek and D. J. Inman, *Symmetric inverse eigenvalue vibration problem and its applications*, Mech. Systems Signal Process., 15 (2001), pp. 11–29.

[18] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[19] D. C. Zimmerman and M. Widengren, *Correcting finite element models using a symmetric eigenstructure assignment technique*, AIAA J., 28 (1990), pp. 1670–1676.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 1, pp. 54–66

# THE MATRIX GOLDEN MEAN AND ITS APPLICATIONS TO RICCATI MATRIX EQUATIONS[*]

YONGDO LIM[†]

**Abstract.** In this paper we generalize the concept of the golden mean of positive numbers to the golden mean of positive definite matrices and apply it to some Riccati algebraic and differential matrix equations. We describe the unique positive definite solutions of the Riccati matrix equations $XA^{-1}X \pm X - (B - A) = 0$ with $0 < A \leq B$ in terms of geometric and golden means of positive definite matrices as well as the asymptotic behavior of the associated Riccati differential equations $\dot{X} = -XA^{-1}X \mp X + (B - A)$. We describe (apparently new) results related to matrix continued fractions, symplectic Hamiltonian matrices, and other matrix means obtained from golden mean–related inequalities via canonical iterative processes.

**Key words.** positive definite matrix, Riccati (differential) matrix equation, geometric mean, golden mean, continued fraction, symplectic Hamiltonian, Riemannian metric

**AMS subject classifications.** 15A24, 93D20, 53B21

**DOI.** 10.1137/050645026

**1. Introduction.** The geometric mean $A\#B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}$ of positive definite matrices $A$ and $B$ appears in the literature with many applications in matrix inequalities (arithmetic-geometric-harmonic mean inequalities [4], [5], [6], [29]), inverse mean problems [2], [3], [20], [37], semidefinite programming (scaling point [47], [48], [22]), geometry (geodesic middle [30], [10], [11], [16], [32], [36]), statistical shape analysis (intrinsic mean [44], [45], [46]), and symmetric matrix word equations [8], [27], [25], [34]. The most important property of the geometric mean is that it has a Riccati matrix equation as the defining equation: The geometric mean $A\#B$ is the unique positive definite solution of the Riccati matrix equation $XA^{-1}X = B$. It also appears as an attracting fixed point of the associated Riccati differential equation $\dot{X} = -XA^{-1}X + B$ on the open convex cone of positive definite matrices [32, Theorem 5.1]: If $X(t)$ is a solution of the differential equation with initial point being any positive definite matrix, then $\lim_{t\to\infty} X(t) = A\#B$.

In this paper we introduce a new matrix mean having characteristics similar to the geometric mean in Riccati matrix equations. The well-known golden ratio or golden section $\frac{1}{2}(1 + \sqrt{5})$, which has been used extensively in art and architecture (cf. [23], [24], [41]), is regarded as the unique positive real solution of the quadratic equation $x^2 - x - 1 = 0$. More generally, the quadratic equations

$$\frac{x^2}{a} \mp x - (b - a) = 0, \quad 0 < a \leq b, \tag{1.1}$$

have the unique positive real solutions $a\natural b := \frac{1}{2}(a + \sqrt{4ab - 3a^2})$, $a\bar{\natural} b := \frac{1}{2}(-a + \sqrt{4ab - 3a^2})$, respectively, realizing the golden ratio as $1\natural 2 = \frac{1}{2}(1+\sqrt{5})$. We consider a

---

[*] Received by the editors November 12, 2005; accepted for publication (in revised form) by R. Bhatia May 15, 2006; published electronically December 21, 2006. This work was supported by grant R01-2006-000-10211-0 from the Basic Research Program of the Korea Science and Engineering Foundation.

http://www.siam.org/journals/simax/29-1/64502.html

[†] Department of Mathematics, Kyungpook National University, Taegu 702-701, Korea (ylim@knu.ac.kr).

natural matrix generalization of (1.1), the Riccati matrix equations under $0 < A \leq B$,

$$XA^{-1}X - X - (B - A) = 0, \tag{1.2}$$

$$XA^{-1}X + X - (B - A) = 0, \tag{1.3}$$

and the associated Riccati matrix differential equations,

$$\dot{X} = -XA^{-1}X + X + (B - A), \tag{1.4}$$

$$\dot{X} = -XA^{-1}X - X + (B - A). \tag{1.5}$$

The classical Riccati matrix (respectively, differential) equation appearing in linear quadratic problems is of the form $XAX - MX - XM^T - B = 0$ and $\dot{X}(t) = B(t) + M(t)X(t) + X(t)M(t)^T - X(t)A(t)X(t)$, where $A \geq 0$ and $B \geq 0$ and $M$ is a square matrix [9], [15], [28]. However, the Riccati equation (1.2) is equivalent to the well-known nonlinear matrix equation $X = Q + NX^{-1}N^T, N > 0$ (see [19], [26], [40]).

In this paper we show that the matrix generalizations of $a\natural b$ and $a\bar{\natural}b$ defined by $A\natural B = \frac{1}{2}(A + A\#(4B - 3A))$ and $A\bar{\natural}B = \frac{1}{2}(-A + A\#(4B - 3A))$ are indeed unique positive definite solutions of (1.2) and (1.3), respectively. The golden mean $A\natural B$ of positive definite matrices with $A \leq B$ is studied in detail in the context of matrix means with close connections to continued fractions and symplectic Hamiltonians. We also prove that for $A < B$, $A\natural B$ and $A\bar{\natural}B$ are attracting fixed points of the differential Riccati equations of (1.4) and (1.5): If $X(t)$ and $Y(t)$ are solutions of the Riccati differential equation (1.4) and (1.5), respectively, with initial value any positive definite matrix, then $A\natural B = \lim_{t\to\infty} X(t)$, $A\bar{\natural}B = \lim_{t\to\infty} Y(t)$, and $\lim_{t\to\infty} X(t)\#Y(t) = A\#(B - A)$.

The paper is organized as follows. In section 2 we briefly review the Riemannian structure and the geometric mean operation on the positive definite convex cone. In section 3 we describe explicit solutions for nonlinear matrix equations which are uniquely equivalent to (1.2) and (1.3). Some basic properties for the golden mean $A\natural B$ of positive definite matrices $A$ and $B$ are found in section 4. The harmonic-geometric-golden mean inequalities are established and corresponding matrix means are obtained from the canonical iterative processes (Theorem 4.3). In section 5 we introduce continued fractions of positive definite matrices and then represent $A\natural B$ as a fixed point of the linear fractional action of a symplectic Hamiltonian matrix and as a limit of repeated continued fractions of positive definite matrices. This shows that for $A < B$, the differential Riccati equations (1.4) and (1.5) have attracting fixed points $A\natural B$ and $A\bar{\natural}B$, respectively. The proof depends heavily on Bougerol's contraction result on symplectic Hamiltonians for the Riemannian metric on the convex cone of positive definite matrices [14].

Throughout this paper we assume that all matrices are square matrices with real entries. Let $\mathrm{Sym}(n, \mathbb{R})$ be the vector space of all $n \times n$ symmetric matrices. For $A \in \mathrm{Sym}(n, \mathbb{R})$, we recall that $A$ is positive semidefinite, denoted by $0 \leq A$, if $\langle x, Ax\rangle \geq 0$ for all $x \in \mathbb{R}^n$, where $\langle \cdot, \cdot \rangle$ denotes the usual inner product on $\mathbb{R}^n$. Similarly, $A$ is positive definite, denoted $0 < A$, if it is positive semidefinite and invertible, or, equivalently, $\langle x, Ax\rangle > 0$ for all nonzero $x$. We denote the set of positive definite (respectively, semidefinite) matrices by $\mathrm{Sym}(n, \mathbb{R})^{++}$ (respectively, $\mathrm{Sym}(n, \mathbb{R})^{+}$).

**2. The geometric mean and invariant metrics.** The geometric mean $A\#B$ of positive semidefinite matrices $A$ and $B$ is defined (and characterized) by the maximum of all $X \geq 0$ for which $\left(\begin{smallmatrix} A & X \\ X & B \end{smallmatrix}\right)$ is positive semidefinite [4]. If $A$ is invertible, then $A\#B = A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2}$. Various alternative characterizations of the geometric mean can be found in matrix inequalities, semidefinite programming, geometry, statistical shape analysis, and matrix word equations. See [4], [5], [10], [11], [7], [20], [32], [36], [37], [44], [45], [46], [47], and see also [6] for multivariable geometric means.

LEMMA 2.1 (Riccati lemma). [illegible] $A$ [illegible] $B$ [illegible] $A\#B$ [illegible] $XA^{-1}X = B$. [illegible]

(i) $A\#B = B\#A$.
(ii) $(A\#B)^{-1} = A^{-1}\#B^{-1}$ [illegible] $A$ [illegible] $B$.
(iii) $M(A\#B)M^T = (MAM^T)\#(MBM^T)$ [illegible] $M$.
(iv) $2(A^{-1}+B^{-1})^{-1} \leq A\#B \leq \frac{1}{2}(A+B)$ [illegible] $A, B$.

[illegible] See [5] or [32]. □

It is known that the open convex cone $\mathrm{Sym}(n,\mathbb{R})^{++}$ of positive definite matrices admits a natural Riemannian structure induced by the trace metric $\langle X, Y\rangle_A := \mathrm{tr}(A^{-1}XA^{-1}Y), A > 0, X, Y \in \mathrm{Sym}(n,\mathbb{R})$. The Riemannian metric distance $\delta(A,B)$ is given by $\delta(A,B) = \left(\sum_{i=1}^{n} \log^2 \lambda_i(A^{-1/2}BA^{-1/2})\right)^{1/2}$, where $\lambda_i(A^{-1/2}BA^{-1/2})$ denote the eigenvalues of $A^{-1/2}BA^{-1/2}$ (see [30]). It is invariant under the matrix inversion and congruence transformations:

$$\delta(A^{-1}, B^{-1}) = \delta(A,B) = \delta(M^TAM, M^TBM), \quad M \in \mathrm{GL}(n,\mathbb{R}). \tag{2.1}$$

In particular, the geometric mean $A\#B$ of positive definite matrices $A$ and $B$ is the unique midpoint (geodesic middle) of $A$ and $B$ for the distance $\delta$ (cf. [30], [32], [11]). The nonpositive curvature property of the Riemannian manifold $\mathrm{Sym}(n,\mathbb{R})^{++}$ is known as the following (equivalent) inequality:

$$\delta(A^t, B^t) \leq t\delta(A,B), \quad 0 \leq t \leq 1. \tag{2.2}$$

In particular, the square root function $X \mapsto X^{1/2}$ is a strict contraction on $\mathrm{Sym}(n,\mathbb{R})^{++}$.

It has recently been proved by Bhatia [10] that the nonpositive curvature property (2.2) holds for metrics inherited from symmetric gauge functions: If $\Phi$ is a symmetric gauge function, for instance one of the Schatten $p$-norms $||x||_p = (\sum |x_i|^p)^{1/p}$, then it defines a natural unitary invariant norm on the complex matrices $||A||_\Phi = \Phi(s_1(A), \ldots, s_n(A))$, where $s_i(A)$ are the singular values of $A$ in nonincreasing order. The corresponding metric distance on the positive definite cone $\Omega$ is determined by $\delta_\Phi(A,B) = ||\log(A^{-1/2}BA^{-1/2})||_\Phi$. The metric $\delta_\Phi$ satisfies the nonpositive curvature property (2.2). We note that $\delta(A,B) = ||\log(A^{-1/2}BA^{-1/2})||_2$.

**3. Riccati matrix equations.** The nonlinear matrix equations $X = Q \pm A^*X^{-1}A$ are extensively studied (see [17], [18], [19], [26], [21], [40], [43]), where $Q > 0$ and $A$ is an $n \times n$ matrix. For positive definite $A$, we have an explicit formula for the positive definite solution of $X = Q + A^*X^{-1}A$.

THEOREM 3.1. [illegible] $A$ [illegible] $B$ [illegible]

$$\begin{aligned} 0 &= X^2 \pm X - A^2, \\ 0 &= BX^{-1}B - X \pm A, \\ 0 &= XA^{-1}X \pm X - B \end{aligned}$$

[illegible]

$$S_1^{\pm}(A) = \frac{1}{2}(\mp I + I\#(I + 4A^2)),$$
$$S_2^{\pm}(A,B) = \frac{1}{2}(\pm A + A\#(A + 4BA^{-1}B)),$$
$$S_3^{\pm}(A,B) = \frac{1}{2}(\mp A + A\#(A + 4B)),$$

[illegible]

[illegible] (1) It is easy to see that $S_1^{\pm}(A) = \frac{1}{2}(\mp I + I\#(I+4A^2))$ are positive definite. By direct computation, $S_1^{\pm}(A)$ are solutions of $X^2 \pm X - A^2 = 0$. Suppose that $X$ is a positive definite solution of $X^2+X-A^2 = 0$. Since $X^2+X-A^2 = (X+\frac{1}{2}I)^2 - A^2 - \frac{1}{4}I$, we have $X + \frac{1}{2}I = (A^2 + \frac{1}{4}I)^{1/2}$ and thus $X = S_1^{+}(A)$.

Next, suppose that $X$ is a positive definite solution of $X^2 - X - A^2 = 0$. Then $(X-\frac{1}{2}I)^2 = A^2 + \frac{1}{4}I$. One can see that $X$ and $A$ commute because $X^2 - X = A^2$. Then by diagonalizing $A$ and $X$, we have $X \geq I$ and $X - \frac{1}{2}I = (A^2 + \frac{1}{4}I)^{1/2}$. Therefore, $X = \frac{1}{2}I + I\#(A^2 + \frac{1}{4}I) = S_1^{-}(A)$.

(2) Consider the matrix equations $X = \pm A + BX^{-1}B$. Setting $Y = A^{-1/2}XA^{-1/2}$ and $D = A^{-1/2}BA^{-1/2}$ we have

$$Y = \pm I + DY^{-1}D, \tag{3.1}$$

respectively. By Lemma 2.1, if $Y$ is a positive definite solution of (3.1), then $Y$ satisfies $D = Y\#(Y \mp I) = (Y^2 \mp Y)^{1/2}$ or

$$Y^2 \mp Y - D^2 = 0. \tag{3.2}$$

Conversely, if $Y$ is a positive definite solution of (3.2), then $Y$ and $D$ commute and hence $Y$ satisfies (3.1). Therefore (3.2) and (3.1) are equivalent, respectively. Solving $Y$ we then have $Y = S_1^{\mp}(D)$ by (1) and therefore

$$\begin{aligned} S_2^{\pm}(A,B) &= A^{1/2}YA^{1/2} = A^{1/2}S_1^{\mp}(A^{-1/2}BA^{-1/2})A^{1/2} \\ &= A^{1/2}\Big(\frac{1}{2}(\pm I + I\#(I + 4(A^{-1/2}BA^{-1/2})^2))\Big)A^{1/2} \\ &= \frac{1}{2}(\pm A + A\#(A + 4BA^{-1}B)), \end{aligned}$$

where the last equality follows from the congruence transformation property of the geometric mean (Lemma 2.1 (iii)).

(3) We consider the matrix equations $XA^{-1}X \pm X - B = 0$. Let $Y = A^{-1/2}XA^{-1/2}$ and $D = A^{-1/2}BA^{-1/2}$. Then $Y^2 \pm Y - D = 0$ and hence $Y = S_1^{\pm}(D^{1/2})$; thus

$$\begin{aligned} S_3^{\pm}(A,B) &= A^{1/2}YA^{1/2} = A^{1/2}S_1^{\pm}(D^{1/2})A^{1/2} \\ &= A^{1/2}\Big(\frac{1}{2}(\mp I + I\#(I + 4D))\Big)A^{1/2} = \frac{1}{2}(\mp A + A\#(A + 4B)). \qquad \square \end{aligned}$$

Corollary 3.2. [illegible] $A$ [illegible] $B$ [illegible] $A \leq B$.

(i) [illegible] $XA^{-1}X - X - (B - A) = 0$ [illegible] $X = A\natural B := \frac{1}{2}(A + A\#(4B - 3A))$.

(ii) [illegible] $XA^{-1}X + X - (B - A) = 0$ [illegible] $X = A\bar{\natural}B := \frac{1}{2}(-A + A\#(4B - 3A))$.

[illegible] 3.3. The positive definite matrix $A\natural B = \frac{1}{2}(A + A\#(4B - 3A))$ is the unique positive definite fixed point of the strict contraction $f(X) = A\#(X + (B - A))$ defined on the positive definite convex cone $\mathrm{Sym}(n, \mathbb{R})^{++}$ for the Riemannian metric distance.

Indeed, the Riccati matrix equation $XA^{-1}X = X + (B - A)$ can be written as $X = A\#(X + (B - A))$ by the Riccati lemma. That is, the positive definite solution of the Riccati matrix equation is a fixed point of the function $f(X) = A\#(X + (B - A))$. Now, the square root function $X \to X^{1/2} = I\#X$ is a strict contraction for the Riemannian metric (2.2), and hence $X \mapsto A\#X$ is a strict contraction by invariance property of the metric. Furthermore, it turns out that every translation $X \mapsto X + C$, $C \geq 0$, is contracted by the Riemannian metric (Proposition 1.6 of [14]). Therefore the map $f(X) = A\#(X + (B - A))$ is a strict contraction defined on the Riemannian manifold $\mathrm{Sym}(n, \mathbb{R})^{++}$ and hence has a unique positive definite fixed point.

[illegible] 3.4. We observe that $A\bar{\natural}B = A\natural B - A$. This can be alternatively explained via a fixed point of a strict contraction. The Riccati matrix equation $XA^{-1}X + X - (B - A) = 0$ is equivalent to $X + A = A\#(X + B)$ by the Riccati lemma; $X + B = (X + A)A^{-1}(X + A) = XA^{-1}X + 2X + A$. That is, $A\bar{\natural}B$ is the unique positive solution of $X + A = A\#(X + B)$. Setting $Y = X + A$, we have $Y = X + A = A\#(X + B) = A\#(X + A + (B - A)) = A\#(Y + B - A)$. By the preceding remark, $Y = A\natural B$ and hence $A\natural B = A\bar{\natural}B + A$.

**4. The golden mean.** It is easy to see that $aA\natural bA = (a\natural b)A$ for any positive scalars $0 < a \leq b$ and, in particular, $A\natural 2A = \frac{1}{2}(1 + \sqrt{5})A$. We call $A\natural B$ the [illegible] of $A$ and $B$.

LEMMA 4.1. [illegible] $C \geq I$. [illegible] $2C^{1/2} - I \leq (4C - 3I)^{1/2} \leq 2C - I$ [illegible] $C = I$.

[illegible]. Use the spectral decomposition of $C$. □

We list some properties of the golden mean $A\natural B$.

PROPOSITION 4.2. [illegible] $A$ [illegible] $B$ [illegible] $A \leq B$.

(i) $M(A\natural B)M^T = (MAM^T)\natural(MBM^T)$ [illegible] $M(A\bar{\natural}B)M^T = (MAM^T)\bar{\natural}(MBM^T)$ [illegible] $n \times n$ [illegible] $M$.

(ii) $A\natural B = A\#B$ [illegible] $A = B$.

(iii) $A\natural B = \frac{1}{2}A^{1/2}\big(I + (4A^{-1/2}BA^{-1/2} - 3I)^{1/2}\big)A^{1/2}$.

(iv) [illegible] $A < B$, [illegible] $A\natural B = \frac{1}{2}\big(A + (B - A)\#(4A + A(B - A)^{-1}A)\big)$.

(v) ( [illegible] )

$$A \leq 2(A^{-1} + B^{-1})^{-1} \leq A\#B \leq A\natural B \leq B.$$

(vi) [illegible] $B \geq 3A$ ( [illegible] $B \leq 3A$), [illegible] $A\natural B \leq \frac{1}{2}(A + B)$ ( [illegible] $A\natural B \geq \frac{1}{2}(A + B)$)

(vii) $(A\natural B)\#(A\bar{\natural}B) = A\#(B - A)$.

(viii) $A\natural B = A\#(B + A\bar{\natural}B)$ [illegible] $A\bar{\natural}B = A\#(B - A\natural B)$.

[illegible]. (i) It follows by linearity of congruence transformations and invariance of geometric mean (Lemma 2.1 (iii)).

Set $C := A^{-1/2}BA^{-1/2}$. Then $C \geq I$ and hence we can apply Lemma 4.1.

(ii) It follows from the invariance property of the geometric and golden means that $A\#B = A\natural B$ if and only if $C^{1/2} = \frac{1}{2}(I + (4C - 3I)^{1/2})$. By Lemma 4.1, this is exactly the case $C = I$ or, equivalently, $A = B$.

(iii) It follows by (i).

(iv) Suppose that $A < B$. Then $C - I > 0$. By (iii) and Lemma 2.1 (iii),

$$\begin{aligned} A\natural B &= \frac{1}{2}A^{1/2}\Big(I + (4A^{-1/2}BA^{-1/2} - 3I)^{1/2}\Big)A^{1/2} \\ &= \frac{1}{2}A^{1/2}\Big(I + (4C - 3I)^{1/2}\Big)A^{1/2} = \frac{1}{2}A^{1/2}\Big(I + (4(C - I) + I)^{1/2}\Big)A^{1/2} \\ &= \frac{1}{2}A^{1/2}\Big(I + (C - I)\#(4I + (C - I)^{-1})\Big)A^{1/2} \\ &= \frac{1}{2}(A + (B - A)\#(4A + A(B - A)^{-1}A)). \end{aligned}$$

(v) By applying the order reversing property of the matrix inversion and the harmonic-geometric mean inequality, we get $A \leq 2(A^{-1} + B^{-1})^{-1} \leq A\#B$. The geometric and golden mean inequality follows from Lemma 4.1 and by (iii),

$$\begin{aligned} A\#B &= A^{1/2}(A^{-1/2}BA^{-1/2})^{1/2}A^{1/2} = A^{1/2}C^{1/2}A^{1/2} \\ &\overset{L(4.1)}{\leq} A^{1/2}\Big(\frac{1}{2}(I + (4C - 3I)^{1/2})\Big)A^{1/2} \overset{\text{(iii)}}{=} A\natural B \leq B. \end{aligned}$$

(vi) It follows by $C^2 - 4C + 3I = (C - 3I)(C - I)$.

(vii) It follows from $(I\natural C)(I\bar{\natural}C) = \frac{1}{4}(I + (4C - 3I)^{1/2})(-I + (4C - 3I)^{1/2}) = C - I$ and the invariance property of the golden mean (i).

(viii) By direct computation

$$\begin{aligned} (A\natural B)A^{-1}(A\natural B) &= \frac{1}{4}\Big(A + A\#(4B - 3A)\Big)A^{-1}\Big(A + A\#(4B - 3A)\Big) \\ &= \frac{1}{4}\Big(I + A\#(4B - 3A)\cdot A^{-1}\Big)\Big(A + A\#(4B - 3A)\Big) \\ &= \frac{1}{4}(A + 2\cdot A\#(4B - 3A) + 4B - 3A) \\ &= \frac{1}{2}(-A + A\#(4B - 3A)) + B = B + A\bar{\natural}B, \end{aligned}$$

where the third equality follows from the Riccati lemma $(A\#(4B-3A))A^{-1}(A\#(4B-3A)) = 4B-3A$. It then follows by the Riccati lemma $A\natural B = A\#(B+A\bar{\natural}B)$. Similarly, $A\bar{\natural}B = A\#(B - A\natural B)$. □

We denote $\mathcal{G}$ by the graph of the Löwner order $\leq$ on the positive definite cone $\mathcal{G} := \{(A,B)|\ 0 < A \leq B\}$. The functions

$$\begin{aligned} &g : \mathcal{G} \to \mathcal{G}, \quad g(A,B) = (A\#B, A\natural B), \\ &h : \mathcal{G} \to \mathcal{G}, \quad h(A,B) = (2(A^{-1} + B^{-1})^{-1}, A\natural B) \end{aligned}$$

are well-defined by the harmonic-geometric-golden mean inequalities (Proposition 4.2 (v)). We denote $g^n$ and $h^n$ by the $n$th iterate of $g$ and $h$.

THEOREM 4.3. [illegible] $(A,B) \in \mathcal{G}$ [illegible] GGM $(A,B)$ [illegible] HGM$(A,B)$ [illegible]

$$\begin{aligned} &\lim_{n\to\infty} g^n(A,B) = (\text{GGM}(A,B), \text{GGM}(A,B)), \\ &\lim_{n\to\infty} h^n(A,B) = (\text{HGM}(A,B), \text{HGM}(A,B)). \end{aligned}$$

[illegible] HGM$(A,B) \leq$ GGM$(A,B)$ [illegible]

[illegible]. Let $A, B$ be positive definite matrices such that $A \leq B$. Setting $(A_1, B_1) = (A, B), (A_{n+1}, B_{n+1}) = g^n(A, B)$, we have $A_2 = A\#B,\ B_2 = A\natural B, A_{n+1} = A_n\#B_n$, and $B_{n+1} = A_n\natural B_n$. By Proposition 4.2 (v), $A_1 = A \leq A\#B = A_2 \leq A\natural B = B_2 \leq B = B_1$. Hence $A_2 \leq B_2$. Doing this process with $(A_2, B_2)$, we get $A_1 \leq A_2 \leq A_2\#B_2 = A_3 \leq A_2\natural B_2 = B_3 \leq B_2 \leq B_1$ and by induction $A_1 \leq A_n \leq A_n\#B_n = A_{n+1} \leq A_n\natural B_n = B_{n+1} \leq B_n \leq B_1$ for all $n \in \mathbb{N}$. This implies that the sequence $A_n$ (respectively, $B_n$) is increasing (respectively, decreasing) bounded above (respectively, below). Thus they converge, say, $A_\infty = \lim_{n\to\infty} A_n, B_\infty = \lim_{n\to\infty} B_n$. Note that $A_\infty$ and $B_\infty$ are positive definite and that

$$A_\infty = \lim_{n\to\infty} A_{n+1} = \lim_{n\to\infty} A_n\#B_n = (\lim_{n\to\infty} A_n)\#(\lim_{n\to\infty} B_n) = A_\infty\#B_\infty.$$

Applying the Riccati lemma, we get $B_\infty = (A_\infty\#B_\infty)A_\infty^{-1}(A_\infty\#B_\infty) = A_\infty A_\infty^{-1} A_\infty = A_\infty$. This completes the proof for the map $g$.

Using the harmonic-golden mean inequality, one can give a similar proof for the map $h$. Furthermore, the order relation $\mathrm{HGM}(A, B) \leq \mathrm{GGM}(A, B)$ follows from the harmonic-geometric mean inequality. □

We call $\mathrm{GGM}(A, B)$ and $\mathrm{HGM}(A, B)$ the [illegible] and [illegible] of $A$ and $B$, respectively.

[illegible] 4.4. We note that the harmonic-arithmetic mean iteration $(A, B) \mapsto (2(A^{-1}+B^{-1})^{-1}, (A+B)/2)$ starting at $(A_0, B_0)$ has a (common) limit, the geometric mean $A_0\#B_0$ (cf. [32]), and that in the positive real case the limit of the arithmetic-geometric mean iteration $(a, b) \mapsto (\sqrt{ab}, (a+b)/2)$ (called the AGM) can be described in terms of complete elliptic integrals. See [12] and [13] for other mean iterations of positive real numbers. In this context, there may be interest in studying the means $\mathrm{GGM}(A, B), \mathrm{HGM}(A, B)$ and the self map of $\mathcal{G}$ defined by

$$\mathcal{G} \to \mathcal{G}, (A, B) \mapsto (\mathrm{HGM}(A, B), \mathrm{GGM}(A, B))$$

even for positive real numbers. For instance, we compute

$$\mathrm{GGM}(1, 2) = 1.58975858639889\ldots, \quad \mathrm{HGM}(1, 2) = 1.56481880877123\ldots.$$

**5. Matrix continued fractions and differential Riccati equations.** In [14] Bougerol considered the closed subsemigroup

$$\mathcal{H} := \left\{ \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathrm{Sp}(2n, \mathbb{R}) : A \text{ is invertible}, BA^T \geq 0, A^TC \geq 0 \right\}$$

of the symplectic Lie group $\mathrm{Sp}(2n, \mathbb{R})$. Here a symplectic matrix $T = \left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$ of order $2n$ means an invertible matrix satisfying $BA^T$ and $A^TC$ are symmetric and $A^TD - C^TB = I$. The members of $\mathcal{H}$ are called [illegible] (cf. [15]). We denote by $\mathcal{H}_0$ the subset of $\mathcal{H}$ for which $(1, 2)$ and $(2, 1)$ entries are positive definite: $BA^T, A^TC > 0$. We briefly review some results about Hamiltonian matrices. For more detail, see [31], [35], and [33]. The semigroup $\mathcal{H}$ has nonempty interior: $\mathcal{H}_0$ is the interior of $\mathcal{H}$ and is an open dense ideal of $\mathcal{H}$. Let $\mathfrak{sp}(2n, \mathbb{R})$ be the Lie algebra of the symplectic Lie group $\mathrm{Sp}(2n, \mathbb{R})$:

$$\mathfrak{sp}(2n, \mathbb{R}) = \left\{ \begin{pmatrix} A & B \\ C & -A^T \end{pmatrix} : B, C \in \mathrm{Sym}(n, \mathbb{R}), A \in \mathrm{M}(n, \mathbb{R}) \right\}.$$

The tangent Lie wedge of $\mathcal{H}$ defined by $L(\mathcal{H}) := \{X \in \mathfrak{sp}(2n,\mathbb{R}) : \exp(tX) \in \mathcal{H} \text{ for all } t \geq 0\}$ is determined by

$$L(\mathcal{H}) = \left\{ \begin{pmatrix} A & B \\ C & -A^T \end{pmatrix} \in \mathfrak{sp}(2n,\mathbb{R}) : B, C \geq 0 \right\}. \tag{5.1}$$

We further note that the interior of $L(\mathcal{H})$ is $L(\mathcal{H})_0 := \{(\begin{smallmatrix} A & B \\ C & -A^T \end{smallmatrix}) : B, C > 0\}$. It then follows by the ideal property of $\mathcal{H}_0$ and the homeomorphic property of exponential mapping near the zero matrix that

$$\exp(L(\mathcal{H})_0) \subset \mathcal{H}_0. \tag{5.2}$$

In [14] Bougerol proved that each member of $\mathcal{H}$ (respectively, $\mathcal{H}_0$) under the action of linear fractional transformations defined by

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} .X = (AX+B)(CX+D)^{-1} \text{ if } CX+D \text{ is invertible}$$

carries the convex cone $\mathrm{Sym}(n,\mathbb{R})^{++}$ into itself [14, Proposition 1.5] and is a (respectively, strict) contraction with respect to the Riemannian metric $\delta(A,B)$ [14, Theorem 1.7]. We further note that each member of $\mathcal{H}$ (respectively, $\mathcal{H}_0$) is also a (respectively, strict) contraction with respect to the Thompson part metric $d(A,B) = ||\log(A^{-1/2}BA^{-1/2})||_\infty$ induced by the spectral norm with an explicit formula (a Birkhoff contraction formula) for least contraction coefficient [38], [39], [35].

We consider the symplectic Hamiltonian matrix $(\begin{smallmatrix} I+AB & A \\ B & I \end{smallmatrix})$ for positive definite matrices $A$ and $B$. It belongs to $\mathcal{H}_0$ since $A(I+AB)^T = A + ABA > 0$ and $(I+AB)^T B = B + BAB > 0$. Now, applying Bougerol's result, the associated fractional transformation

$$\begin{aligned} \Phi_{A,B}(X) &= \begin{pmatrix} I+AB & A \\ B & I \end{pmatrix} .X = ((I+AB)X + A)(BX+I)^{-1} \\ &= (A(BX+I)+X)(BX+I)^{-1} = A + X(BX+I)^{-1} = A + (B+X^{-1})^{-1} \end{aligned}$$

is strictly contracted by the Riemannian distance, and hence it has a unique positive definite fixed point. Furthermore, the fixed point is the limit of the $n$th iterations with any initial point by the Banach fixed point theorem. We assert that the $n$th iteration $\Phi^n_{A,B}(A)$ forms a continued fraction. The continued fraction $[A_1, A_2, \ldots, A_n]$ of positive definite matrices is naturally defined by

$$[A_1] = A_1, \quad [A_1, A_2, \ldots, A_n] = A_1 + [A_2, \ldots, A_n]^{-1}.$$

Then

$$\Phi^n_{A,B}(A) = [\underbrace{A, B, A, B, \ldots, A, B, A}_{2n+1 \text{ terms}}].$$

Indeed, $\Phi_{A,B}(X) = [A, B, X]$ for any $X > 0$ and by induction

$$\begin{aligned} \Phi^n_{A,B}(A) &= \Phi_{A,B}(\Phi^{n-1}_{A,B}(A)) = [A, B, \Phi^{n-1}_{A,B}(A)] = [A, B, [\underbrace{A, B, \ldots, A}_{2n-1}]] \\ &= A + [\underbrace{B, A, B, \ldots, A}_{2n}]^{-1} = [\underbrace{A, B, A, B \ldots, A, B, A}_{2n+1}]. \end{aligned}$$

Similarly,

$$\Phi^n_{A,B}(A+B^{-1}) = [\underbrace{A, B, A, B, \ldots, A, B}_{2(n+1)\ \text{terms}}].$$

THEOREM 5.1. [illegible] $A$ [illegible] $B$ [illegible]

$$\begin{aligned} A\natural(A+B^{-1}) &= \lim_{n\to\infty}[\underbrace{A, B, A, B, \ldots, A, B, A}_{2n+1}] \\ &= \lim_{n\to\infty}[\underbrace{A, B, A, B, \ldots, A, B}_{2n}]. \end{aligned}$$

[illegible] $0 < A < B$ [illegible]

$$A\natural B = \lim_{n\to\infty}[\underbrace{A, (B-A)^{-1}, A, (B-A)^{-1}, \ldots, A}_{2n+1}]$$

[illegible]

$$A\bar{\natural}B = \lim_{n\to\infty}[\underbrace{(B-A)^{-1}, A, (B-A)^{-1}, \ldots, A}_{2n}]^{-1}.$$

[illegible] By the Banach fixed point theorem and by the above observation, the unique positive definite fixed point of $\Phi_{A,B}$ is the limit of the two $n$th iterations starting the points $A$ and $A+B^{-1}$, respectively:

$$\lim_{n\to\infty}[\underbrace{A, B, A, B, \ldots, A}_{2n+1}] = \lim_{n\to\infty}[\underbrace{A, B, A, B, \ldots, A, B}_{2n}].$$

We will show that $A\natural(A+B^{-1})$ is a fixed point of $\Phi_{A,B}$.

Let $X$ be the positive definite fixed point of $\Phi_{A,B}$. Then it satisfies the continued fraction equation $[A, B, X] = X$ and

$$\begin{aligned} X &= [A, B, X] = A + [B, X]^{-1} = A + (B + X^{-1})^{-1} \\ &= A^{1/2}\Big(I + \Big(A^{1/2}BA^{1/2} + (A^{-1/2}XA^{-1/2})^{-1}\Big)^{-1}\Big)A^{1/2} = A^{1/2}[I, C, D]A^{1/2}, \end{aligned}$$

where $C = A^{1/2}BA^{1/2}, D = A^{-1/2}XA^{-1/2}$. This implies that

$$[I, C, D] = A^{-1/2}[A, B, X]A^{-1/2} = A^{-1/2}XA^{-1/2} = D,$$

that is, $D = A^{-1/2}XA^{-1/2}$ is the unique positive definite solution of the equation of continued fraction $[I, C, Y] = Y$. Since $(C + D^{-1})^{-1} = D - D(D + C^{-1})^{-1}D$ for any positive definite matrices $C$ and $D$ (cf. [1], [49]), $[I, C, D] = D$ implies that

$$D^2 - D - C^{-1} = 0. \tag{5.3}$$

By Theorem 3.1, $D = \frac{1}{2}(I + I\#(I + 4C^{-1})) = I\natural(I + C^{-1})$. Therefore,

$$\begin{aligned} X &= A^{1/2}DA^{1/2} = A^{1/2}(I\natural(I + C^{-1}))A^{1/2} \\ &= A^{1/2}(I\natural(I + A^{-1/2}B^{-1}A^{-1/2}))A^{1/2} = A\natural(A + B^{-1}), \end{aligned}$$

where the last equality follows from the invariance property of the golden mean (Proposition 4.2).

Finally we have a similar result for $A\bar{\natural}B$:

$$\begin{aligned}A\bar{\natural}B &= (A\natural B) - A = \Big(\lim_{n\to\infty}[\underbrace{A, (B-A)^{-1}, A, (B-A)^{-1}, \dots, A}_{2n+1}]\Big) - A \\ &= \Big(A + \lim_{n\to\infty}[\underbrace{(B-A)^{-1}, A, (B-A)^{-1}, \dots, A}_{2n}]^{-1}\Big) - A \\ &= \lim_{n\to\infty}[\underbrace{(B-A)^{-1}, A, (B-A)^{-1}, \dots, A}_{2n}]^{-1}. \qquad \square\end{aligned}$$

We consider the [illegible] of the golden mean $A\natural B$, which is defined by $(B^{-1}\natural A^{-1})^{-1}$.

COROLLARY 5.2. [illegible] $0 < A < B$, [illegible] $A\natural B$ [illegible]

$$(B^{-1}\natural A^{-1})^{-1} = (A^{-1} - B^{-1})^{-1}\bar{\natural}(B + (A^{-1} - B^{-1})^{-1}).$$

[illegible] $(A\natural B)^{-1} = (B-A)^{-1}\bar{\natural}(A^{-1} + (B-A)^{-1})$.

[illegible] Let $X = (A^{-1} - B^{-1})^{-1}$ and $Y = B + (A^{-1} - B^{-1})^{-1}$. Then by Theorem 5.1,

$$(X\bar{\natural}Y)^{-1} = \lim_{n\to\infty}[B^{-1}, (A^{-1} - B^{-1})^{-1}, B^{-1}, \dots, (A^{-1} - B^{-1})^{-1}] = B^{-1}\natural A^{-1}. \qquad \square$$

We consider two Riccati differential equations

$$\dot{X} = -XA^{-1}X + X + (B - A), \tag{5.4}$$

$$\dot{X} = -XA^{-1}X - X + (B - A) \tag{5.5}$$

for positive definite matrices $A$ and $B$ with $A < B$.

Let $T = \left(\begin{smallmatrix} I/2 & B-A \\ A^{-1} & -I/2 \end{smallmatrix}\right)$ and $S = \left(\begin{smallmatrix} -I/2 & B-A \\ A^{-1} & I/2 \end{smallmatrix}\right)$. Then since $A\natural B$ is the positive definite solution of the associated matrix equations of (5.4), the graph of $A\natural B$ is $T$-invariant and hence $\exp tT$-invariant. Indeed,

$$\begin{aligned}\begin{pmatrix} I/2 & B-A \\ A^{-1} & -I/2 \end{pmatrix}\begin{pmatrix} (A\natural B)x \\ x \end{pmatrix} &= \begin{pmatrix} \frac{1}{2}(A\natural B)x + (B-A)x \\ (A^{-1}\cdot A\natural B)x - x/2 \end{pmatrix} \\ &= \begin{pmatrix} (A\natural B)y \\ y \end{pmatrix}, \quad y = (A^{-1}\cdot A\natural B)x - x/2,\end{aligned}$$

where we used the fact $(A\natural B)A^{-1}(A\natural B) = B + A\bar{\natural}B = A\natural B + (B-A)$ (Proposition 4.2 (viii)). Identifying the Möbius action of the symplectic group on symmetric matrices as the linear action on the space of isotropic subspaces of $\mathbb{R}^n \times \mathbb{R}^n$ (cf. section 5 of [32]), we have that $\exp tT.(A\natural B) = A\natural B$ for all $t \in \mathbb{R}$. Similarly, $\exp tS.(A\bar{\natural}B) = A\bar{\natural}B$.

However, from $T, S \in L(\mathcal{H})_0$ and (5.2), one sees that $\exp tT$ and $\exp tS$ belong to $\mathcal{H}_0$ for $t > 0$ and hence they are strict contractions for the Riemannian distance. This implies that $A\natural B$ and $A\bar{\natural}B$ are unique positive definite fixed points of $\exp tT$ and $\exp tS$, respectively.

Let $X_0 > 0$ be an arbitrary initial point of the Riccati differential equation (5.4). Then from a standard fact about differentiable Lie group action (cf. section 5 of [32]) one may see that $X(t) := \exp tT.X_0$ is the solution of (5.4). Putting $K := \sup_{0\le s\le 1}\delta(\exp sT.X_0, X_0)$, where $\delta(A,B) = ||\log(A^{-1/2}BA^{-1/2})||_2$ denotes the Riemannian metric on the positive definite cone $\mathrm{Sym}(n,\mathbb{R})^{++}$, we then have for $t = n + s, s \in [0,1], n = 1,2,\ldots,$

$$\begin{aligned}\delta(\exp tT.X_0, A\natural B) &= \delta(\exp(n+s)T.X_0, A\natural B)\\ &\leq \delta(\exp nT.(\exp sT.X_0), \exp nT.X_0) + \delta(\exp nT.X_0, A\natural B)\\ &\leq \alpha^n \delta(\exp sT.X_0, X_0) + \delta(\exp nT.X_0, A\natural B)\\ &\leq \alpha^n K + \alpha^n \delta(X_0, A\natural B) = \alpha^n(K + \delta(X_0, A\natural B)),\end{aligned}$$

where $\alpha$ denotes the least contraction coefficient of the strict contraction $\exp T$. We conclude that the solution $X(t)$ converges to $A\natural B$ as $t \to \infty$. A similar method can be applied for $A\bar{\natural}B$.

THEOREM 5.3. [illegible] $X(t)$ [illegible] $Y(t)$ [illegible] (5.4) [illegible] (5.5) [illegible]

$$A\natural B = \lim_{t\to\infty} X(t) \quad [illegible] \quad A\bar{\natural}B = \lim_{t\to\infty} Y(t).$$

[illegible] $\lim_{t\to\infty} X(t)\#Y(t) = (A\natural B)\#(A\bar{\natural}B) = A\#(B-A)$.

[illegible] The equality $(A\natural B)\#(A\bar{\natural}B) = A\#(B-A)$ follows from Proposition 4.2 (vii). ☐

[illegible] 5.4. The preceding result holds true under the condition $A < B$ and solutions with initial point at positive definite matrices. In the case that $A \leq B$ but not $A < B$, that is, $A^{-1/2}BA^{-1/2}$ has the eigenvalue 1, then $\exp tT$ is not a strict contraction for $t > 0$, and so it is unclear whether solutions $X(t)$ converge to $A\natural B$.

**Acknowledgment.** The author is very grateful to the referees for valuable comments.

## REFERENCES

[1] W. ANDERSON, G. KLEINDORFER, P. KLEINDORFER, AND M. WOODROOFE, *Consistent estimates of the parameters of a linear system*, Ann. Math. Statist., 40 (1969), pp. 2064–2075.

[2] W. N. ANDERSON, JR., M. E. MAYS, T. D. MORLEY, AND G. E. TRAPP, *The contraharmonic mean of HSD matrices*, SIAM J. Algebra Discrete Methods, 8 (1987), pp. 674–682.

[3] W. N. ANDERSON, JR., AND G. E. TRAPP, *Inverse problems for means of matrices*, SIAM J. Algebra Discrete Methods, 7 (1986), pp. 188–192.

[4] T. ANDO, *Topics on Operator Inequalities*, Lecture Notes, Hokkaido University, Sapporo, Japan, 1978.

[5] T. ANDO, *Concavity of certain maps on positive definite matrices and applications to Hadamard products*, Linear Algebra Appl., 26 (1979), pp. 203–241.

[6] T. ANDO, C.-K. LI, AND R. MATHIAS, *Geometric means*, Linear Algebra Appl., 385 (2004), pp. 305–334.

[7] E. ANDRUCHOW, G. CORACH, AND D. STOJANOFF, *Geometric significance of the Löwner-Heinz inequality*, Proc. Amer. Math. Soc., 128 (1999), pp. 1031–1037.

[8] S. N. ARMSTRONG AND C. R. HILLAR, *A Degree Theoretic Approach to the Solvability of Symmetric Word Equations in Positive Definite Letters*, preprint.

[9] D. S. BERNSTEIN, *Matrix Mathematics*, Princeton University Press, Princeton, NJ, Oxford, UK, 2005.

[10] R. BHATIA, *On the exponential metric increasing property*, Linear Algebra Appl., 375 (2003), pp. 211–220.

[11] R. BHATIA AND J. HOLBROOK, *Riemannian geometry and matrix geometric means*, Linear Algebra Appl., 413 (2006), pp. 594–618.

[12] J. M. BORWEIN AND P. B. BORWEIN, *On the mean iteration* $(a,b) \leftarrow (\frac{a+3b}{4}, \frac{\sqrt{ab}+b}{2})$, Math. Comp., 53 (1989), pp. 311–326.

[13] J. M. BORWEIN, P. B. BORWEIN, AND F. GARVAN, *Hypergeometric analogues of the arithmetic-geometric mean iteration*, Constr. Approx., 9 (1993), pp. 509–523.

[14] P. Bougerol, *Kalman filtering with random coefficients and contractions*, SIAM J. Control Optim., 31 (1993), pp. 942–959.

[15] M. Chu, N. Buono, F. Diele, T. Politi, and S. Ragni, *On the semigroup of standard symplectic matrices and its applications*, Linear Algebra Appl., 389 (2004), pp. 215–225.

[16] G. Corach, H. Porta, and L. Recht, *Geodesics and operator means in the space of positive operators*, Internat. J. Math., 4 (1993), pp. 193–202.

[17] J. C. Engwerda, *On the existence of a positive definite solution of the matrix equation* $X + A^TX^{-1}A = I$, Linear Algebra Appl., 194 (1993), pp. 91–108.

[18] J. C. Engwerda, A. C. M. Ran, and A. L. Rijkeboer, *Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation* $X + A^*X^{-1}A = Q$, Linear Algebra Appl., 186 (1993), pp. 255–275.

[19] A. Ferrante and B. Levy, *Hamiltonian solutions of the equation* $X = Q + NX^{-1}N^*$, Linear Algebra Appl., 247 (1996), pp. 359–373.

[20] M. Fiedler and V. Pták, *A new positive definite geometric mean of two positive definite matrices*, Linear Algebra Appl., 251 (1997), pp. 1–20.

[21] C.-H. Guo and P. Lancaster, *Iterative solution of two matrix equations*, Math. Comput., 68 (1999), pp. 1589–1603.

[22] R. A. Hauser and Y. Lim, *Self-scaled barriers for irreducible symmetric cones*, SIAM J. Optim., 12 (2002), pp. 715–723.

[23] R. Herz-Fischler, *A Mathematical History of Division in Extreme and Mean Ratio*, Wilfrid Laurier University Press, Waterloo, ON, 1987.

[24] R. Herz-Fischler, *The home of golden numberism*, Math. Intelligencer, 27 (2005), pp. 69–71.

[25] C. J. Hillar and C. R. Johnson, *Symmetric word equations in two positive definite letters*, Proc. Amer. Math. Soc., 132 (2004), pp. 945–953.

[26] I. G. Ivanov, V. I. Hasanov, and F. Uhlig, *Improved methods and starting values to solve the matrix equations* $X \pm A^*X^{-1}A = I$ *iteratively*, Math. Comp., 74 (2005), pp. 263–278.

[27] C. R. Johnson and C. J. Hillar, *Eigenvalues of words in two positive definite letters*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 916–928.

[28] V. Jurdjevic, *Geometric Control Theory*, Cambridge Stud. Adv. Math. 51, Cambridge University Press, Cambridge, UK, 1997.

[29] F. Kubo and T. Ando, *Means of positive linear operators*, Math. Ann., 246 (1980), pp. 205–224.

[30] S. Lang, *Fundamentals of Differential Geometry*, Grad. Texts in Math. 191, Springer, New York, 1999.

[31] J. D. Lawson and Y. Lim, *Lie semigroups with triple decompositions*, Pacific J. Math., 192 (2000), pp. 393–412.

[32] J. D. Lawson and Y. Lim, *The geometric mean, matrices, metrics, and more*, Amer. Math. Monthly, 108 (2001), pp. 797–812.

[33] J. D. Lawson and Y. Lim, *The symplectic semigroup and Riccati differential equations*, J. Dyn. Control Syst., 12 (2006), pp. 49–77.

[34] J. D. Lawson and Y. Lim, *Solving symmetric matrix word equations via symmetric space machinery*, Linear Algebra Appl., 414 (2006), pp. 560–569.

[35] Y. Lim, *Birkhoff formula for conformal compressions of symmetric cones*, Amer. J. Math., 125 (2003), pp. 167–182.

[36] Y. Lim, *Best approximation in Riemannian geodesic submanifolds of positive definite matrices*, Canad. J. Math., 56 (2004), pp. 776–793.

[37] Y. Lim, *The inverse mean problem of geometric and contraharmonic means*, Linear Algebra Appl., 408 (2005), pp. 221–229.

[38] C. Liverani and M. P. Wojtkowski, *Generalization of the Hilbert metric to the space of positive definite matrices*, Pacific J. Math., 166 (1994), pp. 339–355.

[39] C. Liverani and M. P. Wojtkowski, *Ergodicity in Hamiltonian systems*, in Dynamics Reported, Dynam. Report. Expositions Dynam. Systems (N.S.) 4, Springer, Berlin, 1995, pp. 130–202.

[40] X. Liu and H. Gao, *On the positive definite solutions of the matrix equations* $X^s \pm A^TX^{-t}A = I_n$, Linear Algebra Appl., 368 (2003), pp. 83–97.

[41] M. Livio, *The Golden Ratio*, Broadway Books, New York, 2002.

[42] H. Maass, *Siegel's Modular Forms and Dirichlet Series*, Lecture Notes in Math. 216, Springer, Heidelberg, 1971.

[43] B. Meini, *Efficient computation of the extreme solutions of* $X + A^*X^{-1}A = Q$ *and* $X - A^*X^{-1}A = Q$, Math. Comp., 71 (2002), pp. 1189–1204.

[44] M. Moakher, *A differential geometric approach to the geometric mean of symmetric positive-definite matrices*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 735–747.

[45] M. Moakher, *On averaging symmetric positive-definite tensors*, submitted.

[46] M. Moakher and P. G. Batchelor, *The symmetric space of positive-definite tensors: From geometry to applications and visualization*, in Visualization and Image Processing of Tensor Fields, J. Weickert and H. Hagen, eds., Springer, Berlin, 2005.

[47] Y.-E. Nesterov and M. J. Todd, *Self-scaled barriers and interior-point methods for convex programming*, Math. Oper. Res., 22 (1997), pp. 1–42.

[48] J. F. Sturm, *Similarity and other spectral relations for symmetric cones*, Linear Algebra Appl., 312 (2000), pp. 135–154.

[49] F. Zhang, *Matrix Theory: Basic Results and Techniques*, Springer, New York, 1999.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 67–81

# A FAST SOLVER FOR HSS REPRESENTATIONS VIA SPARSE MATRICES*

S. CHANDRASEKARAN[†], P. DEWILDE[‡], M. GU[§], W. LYONS[¶], AND T. PALS[†]

**Abstract.** In this paper we present a fast direct solver for certain classes of dense structured linear systems that works by first converting the given dense system to a larger system of block sparse equations and then uses standard sparse direct solvers. The kind of matrix structures that we consider are induced by numerical low rank in the off-diagonal blocks of the matrix and are related to the structures exploited by the fast multipole method (FMM) of Greengard and Rokhlin. The special structure that we exploit in this paper is captured by what we term the hierarchically semiseparable (HSS) representation of a matrix. Numerical experiments indicate that the method is probably backward stable.

**Key words.** fast multipole method, low-rank structures, fast solvers, orthogonal factorizations, hierarchically semiseparable representations, sparse matrices, direct sparse solvers

**AMS subject classifications.** 65F05, 65F50

**DOI.** 10.1137/050639028

## 1. Introduction.

Beginning with the early work of Gohberg, Kailath, and Koltracht [6] and Rokhlin [11], and the introduction of the fast multipole method (FMM) of Greengard and Rokhlin [7], it has become clear that many large matrices that arise in practice have a complex low-rank structure in their submatrices that can be exploited efficiently to speed up matrix algorithms. In particular, such structured matrices arise in the numerical solution of integral equations, as fill-in during Gaussian elimination of sparse matrices that come from the discretization of elliptic PDEs, and in many other applications. In earlier work [2] we introduced techniques to design fast and stable direct solvers for such structured matrices based on an implicit $ULV$ factorization algorithm and a matrix representation that we called hierarchically semiseparable (HSS). In this paper we show that linear systems of equations involving such dense structured matrices can be efficiently converted into a larger sparse system of equations that has an ordering of the unknowns permitting a very efficient direct Gaussian elimination solver to be used. This technique has several advantages. First, it makes it possible to exploit the highly developed sparse direct solver technology to attack dense structured problems. Second, it provides a theoretical tool to study these large dense structured matrices. However, in this paper we just concentrate on showing how this technique can be used to design a fast, stable solver for matrices in HSS form only.

---

[*] Received by the editors August 26, 2005; accepted for publication (in revised form) by D. A. Bin May 30, 2006; published electronically December 21, 2006.
http://www.siam.org/journals/simax/29-1/63902.html

[†] Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 (shiv@ece.ucsb.edu, tpals@engineering.ucsb.edu). The research of these authors was supported in part by NSF grant CCR-0204388.

[‡] Faculty of Electrical Engineering, Delft University of Technology, Delft, The Netherlands (p.dewilde@its.tudelft.nl).

[§] Department of Mathematics, University of California at Berkeley, Berkeley, CA 94704 (mgu@math.berkeley.edu). The research of this author was supported in part by NSF grant CCR-0204388.

[¶] Department of Mathematics, University of California at Santa Barbara, Santa Barbara, CA 93106 (lyons@math.ucsb.edu). The research of this author was supported in part by NSF grant CCR-0204388.

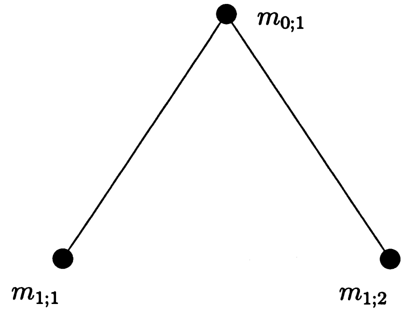


FIG. 1. *One level HSS partition tree with* $m_{0;1} = m_{1;1} + m_{1;2}$.

The idea of explicitly using sparse representations of low-rank structured matrices seems to have first originated in the use of diagonal algebras in time-varying systems theory [5]. Of course, such representations are implicit even in the original FMM papers [7].

## 2. HSS representations.

Usually an $m \times n$ matrix $A$ is represented in terms of its $mn$ entries $A_{i,j}$. The HSS representation of $A$ is another way to present the same information. It tries to exploit the presence of low- (numerical) rank submatrices in $A$. Of course this presumes that we know which submatrices are potentially of low rank. Fortunately, in the application that we have in mind, namely, the numerical solution of elliptic PDEs, this information is usually available. In particular, the HSS representation assumes that the matrix has its low-rank submatrices in the off-diagonal regions. Historically the HSS representation is just a special case of the representations commonly exploited in the FMM literature.

The HSS representation depends directly on a recursive block partitioning of the matrix. It is natural to use a tree to represent these partitions. Suppose at the first level the matrix is partitioned as follows:

$$A = \begin{array}{c} \\ m_{1;1} \\ m_{1;2} \end{array} \begin{array}{c} \begin{array}{cc} m_{1;1} & m_{1;2} \end{array} \\ \begin{pmatrix} A_{1;1,1} & A_{1;1,2} \\ A_{1;2,1} & A_{1;2,2} \end{pmatrix} \end{array}.$$

Then the corresponding HSS partition tree is shown in Figure 1 where it is assumed that $A$ is an $m_{0;1} \times m_{0;1}$ matrix.

The HSS representation tries to exploit the low (numerical) rank of the off-diagonal blocks. The one level HSS tree, for example, is based on the partitioning

$$A = \begin{array}{c} \\ m_{1;1} \\ m_{1;2} \end{array} \begin{array}{c} \begin{array}{cc} m_{1;1} & m_{1;2} \end{array} \\ \begin{pmatrix} D_{1;1} & U_{1;1} B_{1;1,2} V_{1;2}^H \\ U_{1;2} B_{1;2,1} V_{1;1}^H & D_{1;2} \end{pmatrix} \end{array},$$

where clearly the factorization of the off-diagonal blocks can be chosen to be rank-revealing. The tree is shown in Figure 2. At this stage it is not quite clear why, for example, $U_{1;1}$ is written at the first leaf node. One reason is that that particular node corresponds to the first $m_{1;1}$ rows of the matrix, and $U_{1;1}$ is associated with (a portion) of those rows. A better reason will become obvious once we get into section 3. Similarly the expansion coefficient $B_{1;1,2}$ is placed on the edge connecting

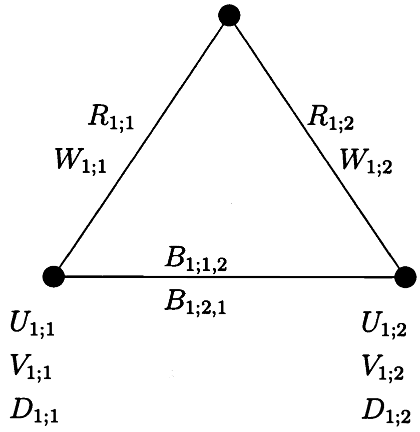


FIG. 2. *One level HSS.*

the two leaves because it sits at the intersection of the rows corresponding to the first leaf node and the columns corresponding to the second leaf node. The matrices $R_{1;i}$ and $W_{1;i}$ have no columns at all and will be explained shortly.

The two level HSS representation is based on the partition

$$A = \begin{array}{c} \\ \begin{pmatrix} m_{2;1} \\ m_{2;2} \end{pmatrix} \\ \begin{pmatrix} m_{2;3} \\ m_{2;4} \end{pmatrix} \end{array} \begin{array}{c} \begin{array}{cc} \begin{pmatrix} m_{2;1} & m_{2;2} \end{pmatrix} & \begin{pmatrix} m_{2;3} & m_{2;4} \end{pmatrix} \end{array} \\ \begin{pmatrix} \begin{pmatrix} A_{2;1,1} & A_{2;1,2} \\ A_{2;2,1} & A_{2;2,2} \end{pmatrix} & A_{1;1,2} \\ A_{1;2,1} & \begin{pmatrix} A_{2;3,3} & A_{2;3,4} \\ A_{2;4,3} & A_{2;4,4} \end{pmatrix} \end{pmatrix} \end{array},$$

where $m_{1;i} = m_{2;2i-1} + m_{2;2i}$ for $i = 1, 2$. The matrices that make up the two level HSS form of $A$ are in turn inferred from the equation

$$A = \begin{pmatrix} \begin{pmatrix} D_{2;1} & U_{2;1}B_{2;1,2}V_{2;2}^H \\ U_{2;2}B_{2;2,1}V_{2;1}^H & D_{2;2} \end{pmatrix} & U_{1;1}B_{1;1,2}V_{1;2}^H \\ U_{1;2}B_{1;2,1}V_{1;1}^H & \begin{pmatrix} D_{2;3} & U_{2;3}B_{2;3,4}V_{2;4}^H \\ U_{2;4}B_{2;4,3}V_{2;3}^H & D_{2;4} \end{pmatrix} \end{pmatrix}.$$

However, the matrices $U_{1;i}$ and $V_{1;i}$ are not part of the two level HSS representation. And, equally importantly, $U_{2;i}$ ($V_{2;i}$) is not chosen as a column (row) basis for $A_{2;i,j}$ ($A_{2;j,i}$). Rather we define [illegible] operators $R_{2;i}$ and $W_{2;i}$ such that

$$U_{1;i} = \begin{pmatrix} U_{2;2i-1}R_{2;2i-1} \\ U_{2;2i}R_{2;2i} \end{pmatrix}, \qquad i = 1, 2, \tag{1}$$

$$V_{1;i} = \begin{pmatrix} V_{2;2i-1}W_{2;2i-1} \\ V_{2;2i}W_{2;2i} \end{pmatrix}, \qquad i = 1, 2. \tag{2}$$

Notice that for this to be possible we must choose $U_{2;i}$ such that it forms a column basis for the submatrix

$$\begin{pmatrix} A_{2;i,1} & \cdots A_{2;i,i-1} & A_{2;i,i+1} & \cdots & A_{2;i,4} \end{pmatrix}.$$

Notice that we obtain the above matrix by taking the $i$th block row from the second level partition of $A$ and dropping the diagonal block $A_{2;i,i}$. Similarly we choose $V_{2;i}$

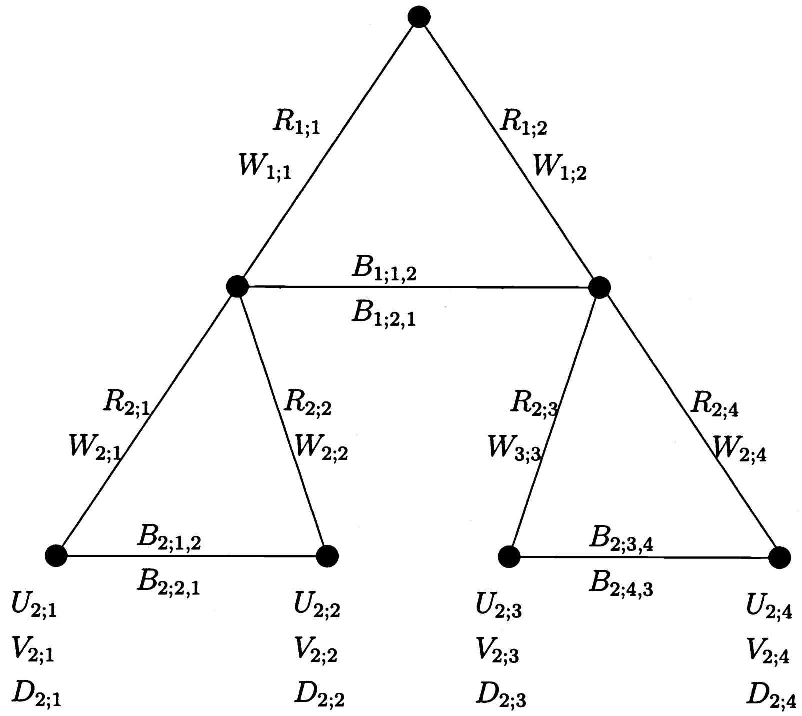


FIG. 3. *Two level HSS.*

to be a row basis for the submatrix

$$\begin{pmatrix} A_{2;1,i} \\ \vdots \\ A_{2;i-1,i} \\ A_{2;i+1,i} \\ \vdots \\ A_{2;4,i} \end{pmatrix}.$$

The two level HSS tree is shown in Figure 3. The translation operators are placed at the edges from the leaves of the second level to their parents in the first level to reflect (1) and (2). The translation operators $R_{1;i}$ and $W_{1;i}$ are not important and are usually chosen to be matrices with no columns at all.

Generally speaking, an HSS representation is a finite binary tree of the type shown in Figure 3, where the dimensions of the matrices at the nodes and leaves must be chosen according to the restrictions that these assemblies

$$\begin{pmatrix} B_{k;2i-1,2i} & R^H_{k+1;2i-1} \\ W_{k+1;2i} & 0 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} B_{k;2i,2i-1} & R^H_{k+1;2i} \\ W_{k+1;2i-1} & 0 \end{pmatrix}$$

are possible if the node $(k,i)$ is not a leaf,[1] and if it is a leaf, then the assembly

$$\begin{pmatrix} D_{k;i} & U_{k;i} \\ V^H_{k;i} & 0 \end{pmatrix}$$

---

[1] The node $(k,i)$ is the $i$th node counting from the left at level $k$ in the tree.

and the multiplications

$$U_{k;i}R_{k;i} \quad \text{and} \quad V_{k;i}W_{k;i}$$

must be possible. In this paper we always assume that $R_{1;i}$ and $W_{1;i}$ have no columns at all.

Given an arbitrary HSS tree and an arbitrary matrix $A$ with the right number of rows and columns, one can [illegible] find an HSS representation for $A$ conforming with the HSS tree. The $O(n^2)$ flops algorithm to carry this out is presented in [2].

**3. Fast multiplication.** The key to the fast inversion algorithm is the fast algorithm for multiplying a matrix in HSS form with a vector. In particular, it is the recursions for the multiplication algorithm that are the key. The recursions we present are exactly the same as those used in the FMM [7].

To be concrete assume that the HSS form of the matrix $A$ is available and that we want to multiply it rapidly with the vector $x$ to obtain $Ax = b$. Of course one method is to first get the componentwise entries $A_{i,j}$ of $A$ and to then use a conventional algorithm. However, that would not be the most efficient thing to do.

Rather, we first observe that we need to multiply submatrices of $x$ with $V_{k;i}$ for each node in the HSS tree. Of course some of these $V_{k;i}$'s are not directly available, namely, those on the nonleaf nodes, but we can get around that using the translation operators $W_{k;i}$. Before we get into the details we need some notation. We will assume that $x_{k;i}$ denotes a submatrix of $x$ partitioned according to the $k$th level of the HSS tree. That is,

$$x = \begin{array}{c} m_{1;1} \\ m_{1;2} \end{array} \begin{pmatrix} x_{1;1} \\ x_{1;2} \end{pmatrix},$$

and

$$x = \begin{array}{c} m_{2;1} \\ m_{2;2} \\ m_{2;3} \\ m_{2;4} \end{array} \begin{pmatrix} x_{2;1} \\ x_{2;2} \\ x_{2;3} \\ x_{2;4} \end{pmatrix},$$

and so on.

Now we observe that at the leaf node $(k, i)$ we can compute

$$g_{k;i} = V_{k;i}^H x_{k;i}.$$

If $(k, i)$ is not a leaf node we can infer

$$\begin{aligned} g_{k;i} &= V_{k;i}^H x_{k;i} \\ &= \begin{pmatrix} V_{k+1;2i-1}W_{k+1;2i-1} \\ V_{k+1;2i}W_{k+1;2i} \end{pmatrix}^H \begin{pmatrix} x_{k+1;2i-1} \\ x_{k+1;2i} \end{pmatrix} \\ &= W_{k+1;2i-1}^H V_{k+1;2i-1}^H x_{k+1;2i-1} + W_{k+1;2i}^H V_{k+1;2i}^H x_{k+1;2i} \\ &= W_{k+1;2i-1}^H g_{k+1;2i-1} + W_{k+1;2i}^H g_{k+1;2i}. \end{aligned}$$

We see therefore that $g_{k;i} = V_{k;i}^H x_{k;i}$ can be computed at each node of the HSS tree very efficiently via the set of equations

$$g_{k;i} = V_{k;i}^H x_{k;i} \qquad \text{at a leaf,} \tag{3}$$

$$= W_{k+1;2i-1}^H g_{k+1;2i-1} + W_{k+1;2i}^H g_{k+1;2i} \qquad \text{at a nonleaf node.} \tag{4}$$

To complete the multiplication let us look in detail at $b_{2;1}$ for a two level HSS tree

$$b_{2;1} = D_{2;1}x_{2;1} + U_{2;1}B_{2;1,2}g_{2;2} + U_{2;1}R_{2;1}B_{1;1,2}g_{1;2},$$

which we can regroup more carefully as follows:

$$b_{2;1} = D_{2;1}x_{2;1} + U_{2;1}(B_{2;1,2}g_{2;2} + R_{2;1}B_{1;1,2}g_{1;2}).$$

This suggests that we define the auxiliary variables $f_{k;i}$ such that

$$b_{k;i} = A_{k;i,i}x_{k;i} + U_{k;i}f_{k;i}.$$

Of course if $(k, i)$ is not a leaf, then we will not have access to the diagonal block $A_{k;i,i}$ or $U_{k;i}$. But in that case we see that we can split the equation using the translation operators $R_{k;i}$ as follows:

$$\begin{pmatrix} b_{k+1;2i-1} \\ b_{k+1;2i} \end{pmatrix} = \begin{pmatrix} A_{k+1;2i-1,2i-1} & U_{k+1;2i-1}B_{k+1;2i-1,2i}V_{k+1;2i}^H \\ U_{k+1;2i}B_{k+1;2i,2i-1}V_{k+1;2i-1}^H & A_{k+1;2i,2i} \end{pmatrix}$$
$$\cdot \begin{pmatrix} x_{k+1;2i-1} \\ x_{k+1;2i} \end{pmatrix} + \begin{pmatrix} U_{k+1;2i-1}R_{k+1;2i-1} \\ U_{k+1;2i}R_{k+1;2i} \end{pmatrix} f_{k;i},$$

which simplifies to the pair of equations

$$b_{k+1;2i-1} = A_{k+1;2i-1,2i-1}x_{k+1;2i-1} + U_{k+1;2i-1}\left(B_{k+1;2i-1,2i}g_{k+1;2i} + R_{k+1;2i-1}f_{k;i}\right),$$
$$b_{k+1;2i} = A_{k+1;2i,2i}x_{k+1;2i-1,2i} + U_{k+1;2i}\left(B_{k+1;2i,2i-1}g_{k+1;2i-1} + R_{k+1;2i-}f_{k;i}\right).$$

This does not seem to lead anywhere, but in fact it does tell us that the recursive equations for the auxiliary variables $f_{k;i}$ are

$$f_{k+1;2i-1} = B_{k+1;2i-1,2i}g_{k+1;2i} + R_{k+1;2i-1}f_{k;i}, \tag{5}$$

$$f_{k+1;2i} = B_{k+1;2i,2i-1}g_{k+1;2i-1} + R_{k+1;2i}f_{k;i}. \tag{6}$$

This looks good, but how do we start off the recursion? In other words, what is $f_{0;1}$? Let us look at its defining equation

$$b = Ax = b_{0;1} = A_{0;1}x_{0;1} + U_{0;1}f_{0;1},$$

which implies that

$$f_{0;1} = (\ ), \tag{7}$$

the empty matrix! Of course at the leaf level we can directly compute the outputs from

$$b_{k;i} = D_{k;i}x_{k;i} + U_{k;i}f_{k;i}. \tag{8}$$

With that we have a complete set of efficient recursions for computing $Ax = b$ given $x$ and the HSS form of $A$.

**4. Sparse representation.** We will now make the effort to write the multiplication recursions in a compact form using matrix notation, and without indices. This will turn out to be the key step that can reveal the way to the fast solver.

We first define some block diagonal matrices. Let $\mathbf{D}$ be a block diagonal matrix formed by ordering $D_{k;i}$ in, say, breadth-first order.[2] Similarly we define block diagonal matrices $\mathbf{U}$ and $\mathbf{V}$. For example, for a two level HSS form, we would have

$$\mathbf{U} = \begin{pmatrix} U_{2;1} & & & \\ & U_{2;2} & & \\ & & U_{2;3} & \\ & & & U_{2;4} \end{pmatrix}.$$

We also arrange all the translation operators $R_{k;i}$ in a block diagonal matrix $\mathbf{R}$, in breadth-first order. Note that there is one $R_{k;i}$ per parent node. So $\mathbf{R}$ will be a block diagonal matrix with a potentially different number of diagonal blocks than, say, $\mathbf{U}$. Similarly we define the block diagonal matrix $\mathbf{W}$. For example, for a two level HSS representation we would have

$$\mathbf{W} = \begin{pmatrix} W_{1;1} & & & & & \\ & W_{1;2} & & & & \\ & & W_{2;1} & & & \\ & & & W_{2;2} & & \\ & & & & W_{2;3} & \\ & & & & & W_{2;4} \end{pmatrix}.$$

We also arrange the $B_{k;i,j}$ in a block diagonal matrix $\mathbf{B}$, with the $B_{k;i,j}$ in breadth-first order, and within a node we place $B_{k+1;2i-1,2i}$ before $B_{k+1;2i,2i-1}$. So for a two level HSS form we would have

$$\mathbf{B} = \begin{pmatrix} B_{1;1,2} & & & & & \\ & B_{1;2,1} & & & & \\ & & B_{2;1,2} & & & \\ & & & B_{2;2,1} & & \\ & & & & B_{2;3,4} & \\ & & & & & B_{2;4,3} \end{pmatrix}.$$

We next define the shift-down operator $Z_{\downarrow}$ on trees. Given a binary tree with matrices on each node, the action of $Z_{\downarrow}$ on the binary tree is to produce an identical tree in which the matrix on every parent node has been moved into the children. The matrices at the leaves are dropped off. The root node acquires a zero matrix. For example for a two level HSS tree the action of $Z_{\downarrow}$ (in the depth-first order for input and output) is expressed by the following equation corresponding to Figure 4:

$$\underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}}_{Z_{\downarrow}} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{pmatrix} = \begin{pmatrix} 0 \\ a \\ a \\ b \\ b \\ c \\ c \end{pmatrix}. \tag{9}$$

---

[2] We are free to pick this order, but once we have chosen an order we must stick with it for the remaining diagonal matrices too.

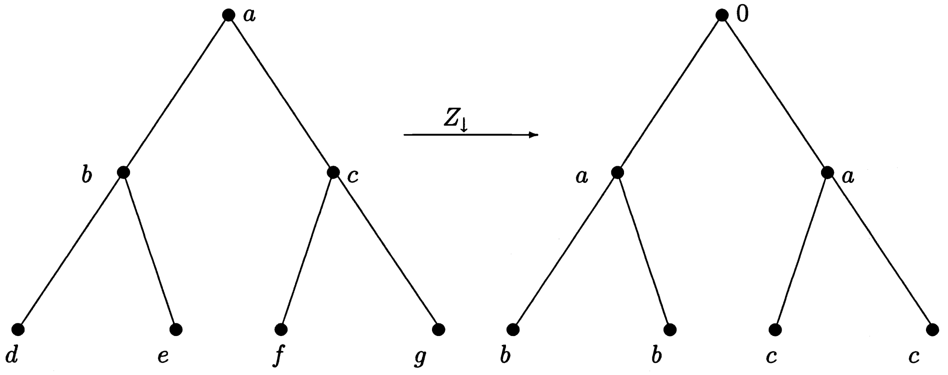


FIG. 4. *Action of $Z_\downarrow$ on a two level HSS tree.*

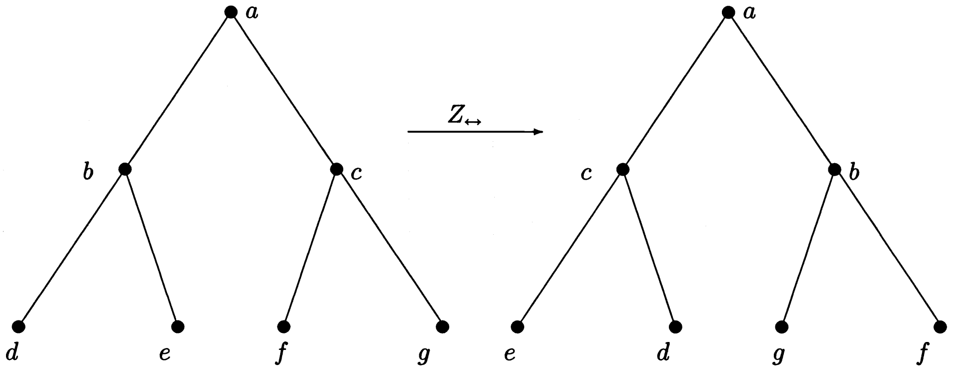


FIG. 5. *Action of $Z_\leftrightarrow$ on a two level HSS tree.*

As can be seen $Z_\downarrow$ is very sparse and noninvertible.

Now we define the twiddle operator $Z_\leftrightarrow$ on trees. When $Z_\leftrightarrow$ acts on a binary tree with matrices on each node, it exchanges the matrices on sibling nodes. The following equation gives an explicit representation for $Z_\leftrightarrow$ on a two level HSS tree (which is shown pictorially in Figure 5):

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}}_{Z_\leftrightarrow} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{pmatrix} = \begin{pmatrix} a \\ c \\ b \\ e \\ d \\ g \\ f \end{pmatrix}.$$

Note that $Z_\leftrightarrow$ is a permutation matrix (which are always very sparse, of course).

Now let us assign the intermediate quantities $g_{k;i}$ and $f_{k;i}$ to the corresponding nodes on the HSS tree. Naturally we can then stack them up in breadth-first ordering in a single block vector and call them $\mathbf{g}$ and $\mathbf{f}$. For example, for the two level HSS tree we would have

$$\mathbf{f} = \begin{pmatrix} f_{0;1} \\ f_{1;1} \\ f_{1;2} \\ f_{2;1} \\ f_{2;2} \\ f_{2;3} \\ f_{2;4} \end{pmatrix}.$$

We also need to define a projection operator $\mathbf{P}_{\text{leaf}}$ that acting on a block vector like $\mathbf{f}$ would return the restriction of it to the leaf nodes. For example, for the two level HSS tree we would have

$$\underbrace{\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\mathbf{P}_{\text{leaf}}} \begin{pmatrix} f_{0;1} \\ f_{1;1} \\ f_{1;2} \\ f_{2;1} \\ f_{2;2} \\ f_{2;3} \\ f_{2;4} \end{pmatrix} = \begin{pmatrix} f_{2;1} \\ f_{2;2} \\ f_{2;3} \\ f_{2;4} \end{pmatrix}. \tag{10}$$

We also define $\mathbf{x}$ and $\mathbf{b}$ as the two block vectors obtained by arranging the submatrices of $x$ and $b$ according to the leaf partitions of the HSS tree. So, for example, in the case of a two level HSS tree we would have

$$\mathbf{x} = \begin{pmatrix} x_{2;1} \\ x_{2;2} \\ x_{2;3} \\ x_{2;4} \end{pmatrix}.$$

Of course this is just $x$ in this case. But if we had ordered the tree nodes (and hence leaves) in some other order this may not have been the case.

With these new matrices we can rewrite the fast multiplication recursions in compact form. Let us start with the pair (3) and (4) which can be written together as

$$\mathbf{g} = \mathbf{P}_{\text{leaf}}^H \mathbf{V}^H \mathbf{x} + Z_{\downarrow}^H \mathbf{W}^H \mathbf{g}. \tag{11}$$

It is very important for the reader to understand why the single equation above is exactly equivalent to the pair (3) and (4). For example, let us check if (3) is captured correctly.

To do that we can apply the leaf projection operator from the left in (11) and obtain

$$\mathbf{P}_{\text{leaf}} \mathbf{g} = \mathbf{P}_{\text{leaf}} \mathbf{P}_{\text{leaf}}^H \mathbf{V}^H \mathbf{x} + \mathbf{P}_{\text{leaf}} Z_{\downarrow}^H \mathbf{W}^H \mathbf{g}. \tag{12}$$

We need to understand the significance of $\mathbf{P}_{\text{leaf}}^H$ and $Z_{\downarrow}^H$ and their relationship to $\mathbf{P}_{\text{leaf}}$.

For example, $\mathbf{P}_{\text{leaf}}^H$ is the pseudoinverse of $\mathbf{P}_{\text{leaf}}$, as $\mathbf{P}_{\text{leaf}}$ is an orthogonal projector

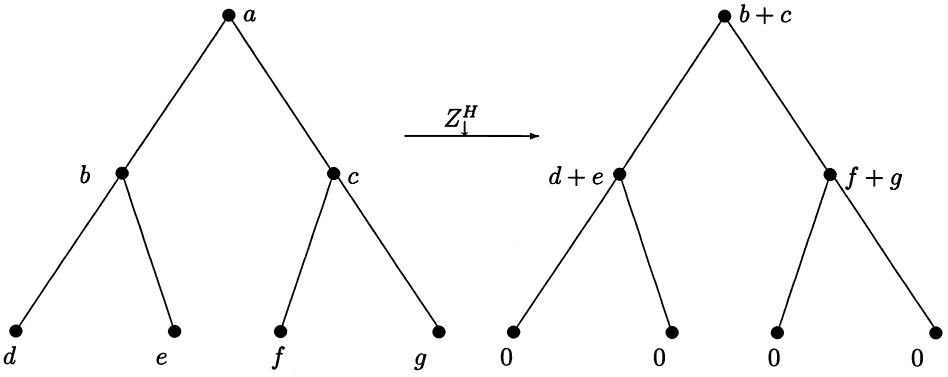


FIG. 6. *Action of $Z_{\downarrow}^{H}$ on a two level HSS tree.*

"onto the leaves of the HSS tree." So if we look at the example in (10) we have

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ f_{2;1} \\ f_{2;2} \\ f_{2;3} \\ f_{2;4} \end{pmatrix} = \mathbf{P}_{\text{leaf}}^{H} \begin{pmatrix} f_{2;1} \\ f_{2;2} \\ f_{2;3} \\ f_{2;4} \end{pmatrix}.$$

From this we can see that $\mathbf{P}_{\text{leaf}}^{H}$ [illegible] a block vector inside an HSS tree with the blocks assigned to the leaves and zeros assigned to the parent nodes. It follows that $\mathbf{P}_{\text{leaf}}\mathbf{P}_{\text{leaf}}^{H} = I$.

Next we look at $Z_{\downarrow}^{H}$. Since the action of $Z_{\downarrow}$ on an HSS tree is to move the vectors at the parent node down into the child nodes, it is not surprising to learn that $Z_{\downarrow}^{H}$ does nearly the opposite; it adds the vectors in the child nodes together and assigns them to the parent nodes, while the leaf nodes are assigned zeros. This is depicted in Figure 6. From this it follows that $\mathbf{P}_{\text{leaf}}Z_{\downarrow}^{H} = 0$.

Putting all this together we see that (12) can be simplified to

$$\mathbf{P}_{\text{leaf}}\mathbf{g} = \mathbf{V}^{H}\mathbf{x},$$

which is exactly (3) written using block matrices.

Next we quickly describe how (4) is embedded in (11). For this we need to consider the nonleaf nodes on both sides of (11). We can do so, for example, by multiplying both sides by $I - \mathbf{P}_{\text{leaf}}^{H}\mathbf{P}_{\text{leaf}}$. The latter acts on an HSS tree by setting all the vectors at the leaf nodes to zero. From this, and our earlier description of $\mathbf{P}_{\text{leaf}}^{H}$ and $Z_{\downarrow}^{H}$, it is easy to verify that $(I - \mathbf{P}_{\text{leaf}}^{H}\mathbf{P}_{\text{leaf}})\mathbf{P}_{\text{leaf}}^{H} = 0$ and $(I - \mathbf{P}_{\text{leaf}}^{H}\mathbf{P}_{\text{leaf}})Z_{\downarrow}^{H} = Z_{\downarrow}^{H}$. Therefore when we multiply both sides of (11) by $(I - \mathbf{P}_{\text{leaf}}^{H}\mathbf{P}_{\text{leaf}})$ we obtain

$$(I - \mathbf{P}_{\text{leaf}}^{H}\mathbf{P}_{\text{leaf}})\mathbf{g} = Z_{\downarrow}^{H}\mathbf{W}^{H}\mathbf{g},$$

which when written out componentwise for each nonleaf node yields (4).

Next we observe that (7), (5), and (6) can be combined and written as the single equation

$$\mathbf{f} = \mathbf{R}Z_{\downarrow}\mathbf{f} + \mathbf{B}Z_{\leftrightarrow}\mathbf{g}. \tag{13}$$

Finally, we can write the output (8) as

$$\mathbf{b} = \mathbf{D}\mathbf{x} + \mathbf{U}\mathbf{P}_{\text{leaf}}\mathbf{f}. \tag{14}$$

It is more convenient to combine the three equations (11), (13), and (14) into the single equation

$$\underbrace{\begin{pmatrix} \mathbf{D} & 0 & \mathbf{U}\mathbf{P}_{\text{leaf}} \\ 0 & \mathbf{B}Z_{\leftrightarrow} & \mathbf{R}Z_{\downarrow} - I \\ \mathbf{P}_{\text{leaf}}^{H}\mathbf{V}^{H} & Z_{\downarrow}^{H}\mathbf{W}^{H} - I & 0 \end{pmatrix}}_{\mathbf{S}} \begin{pmatrix} \mathbf{x} \\ \mathbf{g} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ 0 \\ 0 \end{pmatrix}. \tag{15}$$

We first observe that the matrix $\mathbf{S}$ is extremely sparse. In particular, $\mathbf{S}$ is a block matrix with at most three nonzero blocks in every block row. For example, the first block row of (15) reads

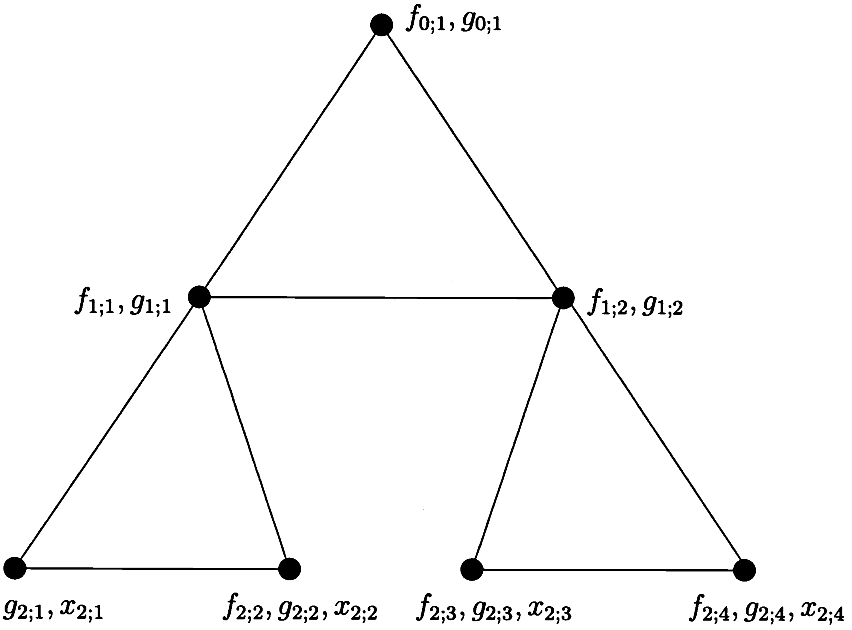
$$D_{2;1}x_{2;1} + U_{2;1}f_{2;1} = b_{2;1},$$

and it shows that $\mathbf{S}$ has only two nonzero blocks, $D_{2;1}$ and $U_{2;1}$, in its first block row. The general observation, that $\mathbf{S}$ has at most three nonzero blocks in any block row, follows from the recursions for the fast multiplication algorithm, (3) to (7).

It is now convenient to look at a graph representation of $\mathbf{S}$. We will use the standard one from text books (see [1, section 6.4.2]). Usually sparse matrices are viewed elementwise and the corresponding graphs have elements on the edges. In our case it is best to view $\mathbf{S}$ as a block sparse matrix and to look at the corresponding graph instead. First we observe that even though $\mathbf{S}$ is not a Hermitian matrix, its nonzero blocks form a structurally symmetric matrix; that is, if the $(i, j)$ block of $\mathbf{S}$ is a structural nonzero block, then the $(j, i)$ block is also a structural nonzero block. Therefore, we can use an undirected graph to represent the block sparsity of $\mathbf{S}$. An example of the graph we will use for a two level HSS form is shown in Figure 7. The graph is set up as follows. First certain block rows and columns are assigned (or associated) with a node of the graph. For example, in Figure 7, the block column corresponding to the unknowns $f_{0;1}$ and $g_{0;1}$ is assigned to the topmost node in the figure. Similarly the block column corresponding to the unknowns $f_{2;1}$, $g_{2;1}$, and $x_{2;1}$ is assigned to the bottom leftmost leaf node. Once a block column has been associated with a node of the graph the corresponding block row is also associated with the same node. Note that the nodes for the graph representation of $\mathbf{S}$ are exactly the nodes of the HSS tree. This is not a coincidence. Once the nodes have been assigned the edges for the graph are picked according to the structural nonzero blocks of $\mathbf{S}$. For example, the equation

$$g_{1;2} - W_{2;3}^{H}g_{2;3} - W_{2;4}^{H}g_{2;4} = 0$$

is one of the rows of the equation expressed by (15). Since this equation connects the block variable $g_{2;3}$ with the block variable $g_{1;2}$ there is an edge in the graph of Figure 7 between the two nodes connecting these two variables. We do a similar thing for every block row equation of (15), drawing an edge between two nodes of the graph if there is a block equation connecting unknowns in the two nodes. The resulting graph representation for $\mathbf{S}$ for a two level HSS form is shown in Figure 7. The assignment of block rows and columns in this figure might seem arbitrary but the intention will become clear soon. Definitely one of the reasons was to make clear that



$f_{0;1}, g_{0;1}$
$f_{1;1}, g_{1;1}$
$f_{1;2}, g_{1;2}$
$f_{2;1}, g_{2;1}, x_{2;1}$
$f_{2;2}, g_{2;2}, x_{2;2}$
$f_{2;3}, g_{2;3}, x_{2;3}$
$f_{2;4}, g_{2;4}, x_{2;4}$

FIG. 7. *Block sparse graph of* **S** *arising from two level HSS form.*

the graph representing the sparsity of **S** is closely related to the graph representing the HSS form of $A$.

Since **S** is very sparse it naturally raises the idea that we could solve the sparse system of equations (15) for $x$ efficiently using a [illegible] sparse solver. However, to establish that we must first establish that the system is invertible (if the original matrix $A$ is) and that we will not incur too much fill-in during Gaussian elimination on **S**.

We begin with the first issue: does $\mathbf{S}^{-1}$ exist whenever $A^{-1}$ exists? While resolving this question we will discover a remarkable diagonal formula for HSS representations. First, observe that the bottom $2 \times 2$ principal submatrix of **S** is invertible with an inverse given by the explicit formula

$$\begin{pmatrix} \mathbf{B}Z_{\leftrightarrow} & \mathbf{R}Z_{\downarrow} - I \\ Z_{\downarrow}^{H}\mathbf{W}^{H} - I & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 0 & (Z_{\downarrow}^{H}\mathbf{W}^{H} - I)^{-1} \\ (\mathbf{R}Z_{\downarrow} - I)^{-1} & -(\mathbf{R}Z_{\downarrow} - I)^{-1}\mathbf{B}Z_{\leftrightarrow}(Z_{\downarrow}^{H}\mathbf{W}^{H} - I)^{-1} \end{pmatrix}.$$

Of course the validity of this formula hinges upon the existence of the two inverses

$$(\mathbf{W}Z_{\downarrow} - I)^{-1} \quad \text{and} \quad (\mathbf{R}Z_{\downarrow} - I)^{-1}.$$

But these two inverses always exist. The reasoning is as follows. We see from (9) that $Z_{\downarrow}$ is nilpotent. Since **W** and **R** are block diagonal matrices with block sizes chosen to be compatible with the block identities in $Z_{\downarrow}$, it follows that $\mathbf{W}Z_{\downarrow}$ and $\mathbf{R}Z_{\downarrow}$ are also nilpotent matrices. From this it follows that the above two inverses always exist. This proves our assertion.

Now we can [illegible] solve (15) for **x** and obtain

$$\left(\mathbf{D} + \mathbf{U}\mathbf{P}_{\text{leaf}}(I - \mathbf{R}Z_{\downarrow})^{-1}\mathbf{B}Z_{\leftrightarrow}(I - Z_{\downarrow}^{H}\mathbf{W}^{H})^{-1}\mathbf{P}_{\text{leaf}}^{H}\mathbf{V}^{H}\right)\mathbf{x} = \mathbf{b}. \tag{16}$$

Since this is true for all $x$, it follows that

$$A = \mathbf{D} + \mathbf{U}\mathbf{P}_{\text{leaf}}(I - \mathbf{R}Z_{\downarrow})^{-1}\mathbf{B}Z_{\leftrightarrow}(I - Z_{\downarrow}^{H}\mathbf{W}^{H})^{-1}\mathbf{P}_{\text{leaf}}^{H}\mathbf{V}^{H}. \tag{17}$$

This is a compact diagonal representation of the HSS form of $A$. It therefore follows that if $A$ is invertible then the sparse matrix $\mathbf{S}$ in (15) is also invertible, since $A$ is just the $(1,1)$ Schur complement of $\mathbf{S}$.

So it is clear that to solve $Ax = b$ for $x$ we could solve the sparse system of equations (15) instead. But to establish that there is a computational advantage in doing so, we must show that the sparse system (15) has an ordering that will not fill in during Gaussian elimination. Of course if we first eliminate $\mathbf{f}$ and then $\mathbf{g}$, we will get exactly $A$, which is the original matrix and completely filled in!

To find a better ordering we look at the block sparse graph for the system of which an example for the two level HSS representation is shown in Figure 7. From that figure it is obvious that there will be no block fill-in for the nested dissection ordering of the unknowns, that is, if we eliminate in the following block order: $(\, f_{2;1} \quad g_{2;1} \quad x_{2;1}\,)$, $(\, f_{2;2} \quad g_{2;2} \quad x_{2;2}\,)$, $(\, f_{2;3} \quad g_{2;3} \quad x_{2;3}\,)$, $(\, f_{2;4} \quad g_{2;4} \quad x_{2;4}\,)$, $(\, f_{1;1} \quad g_{1;1}\,)$, $(\, f_{1;2} \quad g_{1;2}\,)$, $(\, f_{0;1} \quad g_{0;1}\,)$. To see why this is so, note that after eliminating the variables $f_{2;1}$, $g_{2;1}$, and $x_{2;1}$, for example, the remaining equations will have no new nonzero blocks (see [1, sections 6.4.4 and 6.5.3] for further explanations on how determine fill-in during Gaussian elimination from the graph representation).

In general, in the nested dissection ordering all the variables on the left subtree are ordered before all the variables in the right subtree, with the variables on the root node coming last. Of course, the variables in the left and right subtrees are themselves ordered recursively in nested dissection order.

The bottom line is that there [illegible] a no fill-in Gaussian elimination order. But what about pivoting to ensure numerical stability? That is a more complicated question, and we do not answer it here. Rather, we just observe that the block sparse graph also shows that we can get an efficient sparse $QR$ factorization in the nested dissection ordering.

To see this let us follow through the first step of a block Givens $QR$ factorization algorithm on the two level HSS form shown in Figure 7. We first try to eliminate the node containing $f_{2;1}$, $g_{2;1}$, and $x_{2;1}$. We have to first apply a block Givens rotation involving the pivot row and the row corresponding to the variables $f_{2;2}$, $g_{2;2}$, and $x_{2;2}$. We note that the only possible fill-in in this row and the pivot row must correspond to the variables $f_{1;1}$ and $g_{1;1}$. But both these positions are already nonzero, so no fill-in edges have to be added to the graph. If we now proceed with the nested dissection ordering, we can argue similarly that no fill-in edges at all will be added to the block sparse graph (see [1, section 6.6.4]).

Hence, to avoid numerical instabilities, we can just use a sparse $QR$ factorization method. Since there is essentially no fill-in, we obtain a solver that is numerically stable and is linear in the dimension of the matrix $A$, with a constant that depends on the size of the $B_{k;i,j}$ matrices. In particular, the number of flops is a constant times the sum of the cube of the sizes of the $B_{k;i,j}$'s. This can be inferred as follows.

First note that every block equation has at most three block terms. Therefore, every stage of the block $QR$ factorization involves a constant number of matrix multiplications. For the sake of simplicity, let us consider the case when every matrix in the HSS form is no bigger than $p \times p$ for some integer $p$. Then, it is clear that at every stage of the block $QR$ algorithm, the number of flops will not exceed some constant times $p^3$. Since there is no fill-in during the $QR$ factorization, it also follows that the total number of flops will not exceed the number of nodes in the HSS tree times $p^3$ times some constant. But the number of nodes in the HSS tree cannot exceed $n/p$ times some constant, where $n$ is the dimension of the matrix $A$. Therefore, in this case, the number of flops is $O(np^2)$.

We caution that to construct an HSS representation from a dense matrix will in general require $O(n^2)$ flops (using the algorithm presented in [2], for example). The same paper [2] also describes examples where the HSS representation can be computed in $O(n)$ flops. Similarly, the entire FMM literature can be viewed as a repository of examples where the FMM representation can be computed in $O(n)$ flops, and from which the HSS representation in turn can be computed in $O(n)$ flops.

**5. Numerical experiments.** We now describe some numerical experiments that exhibit the efficiency and the stability of the sparse solver approach. All experiments were carried out on a 1GHz PowerPC G4 machine with 1.5GB RAM and a 167MHz bus. We used the vendor supplied BLAS.

The $n \times n$ matrix $A$ was chosen according to the formula $A_{i,j} = \sqrt{|x_i^{(n)} - x_j^{(n)}|}$, with the points $x_i^{(n)} = \cos(\pi(2i+1)/2n)$ as the zeros of the $n$th Chebyshev polynomial. The HSS tree was decided by a standard dyadic division of the interval $[-1, 1]$. The intervals were repeatedly divided in half until there were less than $p$ points left. The value of $p$ was chosen according to the matrix size to enable better memory behavior. Since the zeros of the Chebyshev polynomial cluster at the end points the resulting HSS tree was not uniform. We measure the skewness of the HSS tree as the ratio of the longest path (shortest distance) from a root to a leaf to the shortest path from a root to a leaf. The HSS form of the matrix was computed beforehand using the algorithm in the earlier paper [2] to eight digits of accuracy. It is well known that for matrices of the form we are considering in this experiment the ranks of the $B_{k;i,j}$'s are essentially proportional to the logarithm of the accuracy. Therefore, in this experiment the sizes of the $B_{k;i,j}$'s were essentially constant, independent of the matrix size. Therefore, we should expect the CPU time of the solver to scale linearly in the matrix size.

The experimental data are reported in Table 1. The first column shows that we tried matrices that varied in size from $256 \times 256$ to $131072 \times 131072$. The second column shows the factor $p$ that decides the maximum number of rows (and columns) in a leaf node. The skewness of the HSS tree of the various matrices that we tried is shown in column three. The fourth column shows the CPU time required by the sparse solver.

The sparse solver we used was a custom built block $QR$ solver. We ordered the sparse matrix in (15) in nested dissection order. As can be seen from column four of the table the solver is essentially a linear time solver as predicted by the theory, and that it is not affected adversely by the skewness of the HSS tree.

In column five we show the backward error for each solve. The backward error for solving the system $Ax = b$ with computed solution $\hat{x}$ is defined to be the ratio of the smallest 2-norm of any matrix $E$ that satisfies the equation $(A+E)\hat{x} = b$, and the 2-norm of $A$ (see [1, section 1.4.6]). As can be seen from column five, the backward error for our method is essentially machine precision. This shows that the method behaved in a backward stable manner in this set of experiments.

**6. Conclusion.** We have shown that a fast direct solver for linear systems of equations with the coefficient matrix in HSS form can be easily constructed from a sparse solver. The resulting algorithm is fast and stable.

It is easy to see that this idea can easily be extended to handle more complex

TABLE 1

*Speed and stability of sparse solver for HSS forms.*

| Matrix size | Leaf block size | Skewness of tree | CPU time in seconds | Backward error |
|---|---|---|---|---|
| 256 | 13 | 2 | 0.07 | 9.90765e-17 |
| 512 | 14 | 1.8 | 0.15 | 1.01727e-16 |
| 1024 | 15 | 1.83333 | 0.32 | 2.86709e-16 |
| 2048 | 16 | 1.85714 | 0.68 | 5.5083e-17 |
| 4096 | 17 | 1.875 | 1.43 | 8.24819e-17 |
| 8192 | 18 | 1.88889 | 2.87 | 4.0822e-17 |
| 16384 | 19 | 1.9 | 5.57 | 5.32472e-17 |
| 32768 | 20 | 2 | 11.29 | 4.96643e-17 |
| 65536 | 21 | 2 | 25.43 | 8.64522e-17 |
| 131072 | 22 | 2 | 53.88 | 8.51812e-17 |

partitions of the matrix than the one used in the HSS representation. In particular, the method can easily be extended to handle a full FMM representation of the matrix (see [10]), the hierarchical matrix representation (see [8, 9]), and the sequentially semiseparable representation (see [3, 4]). These matters will be presented in a companion paper.

## REFERENCES

[1] Å. BJÖRCK, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[2] S. CHANDRASEKARAN, M. GU, AND T. PALS, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 603–622.

[3] S. CHANDRASEKARAN, P. DEWILDE, M. GU, T. PALS, X. SUN, A.-J. VAN DER VEEN, AND D. WHITE, *Some fast algorithms for sequentially semiseparable representations*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 341–364.

[4] S. CHANDRASEKARAN, P. DEWILDE, M. GU, T. PALS, AND A. VAN DER VEEN, *Fast stable solver for sequentially semi-separable linear systems of equations*, in HiPC 202, S. Sahni, ed., Lecture Notes in Comput. Sci. 2552, Springer-Verlag, Berlin, 2002, pp. 545–554.

[5] P. DEWILDE AND A. VAN DER VEEN, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[6] I. GOHBERG, T. KAILATH, AND I. KOLTRACHT, *Linear complexity algorithms for semiseparable matrices*, Integral Equations Operator Theory, 8 (1985), pp. 780–804.

[7] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[8] W. HACKBUSCH, *A sparse arithmetic based on $\mathcal{H}$-matrices. Part* I: *Introduction to $\mathcal{H}$-matrices*, Computing, 62 (1999), pp. 89–108.

[9] W. HACKBUSCH, B. N. KHOROMSKIJ, AND S. SAUTER, *On $H^2$-Matrices*, preprint 50, MPI, Leipzig, 1999.

[10] T. PALS, *Multipole for Scattering Computations: Spectral Discretization, Stabilization, Fast Solvers*, Ph.D. thesis, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA, 2004.

[11] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, J. Comput. Phys., 60 (1985), pp. 187–207.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 82–97

# MINIMIZING THE CONDITION NUMBER FOR SMALL RANK MODIFICATIONS*

CHEN GREIF† AND JAMES M. VARAH†

**Abstract.** We consider the problem of minimizing the condition number of a low rank modification of a matrix. Analytical results show that the minimum, which is not necessarily unique, can be obtained and expressed by a small number of eigenpairs or singular pairs. The symmetric and the nonsymmetric cases are analyzed, and numerical experiments illustrate the analytical observations.

**Key words.** condition number, low rank modifications, minimization, interlacing

**AMS subject classifications.** 65F35, 15A12

**DOI.** 10.1137/050647554

## 1. Introduction.

Let $A \in \mathbb{R}^{n\times n}$ be a general matrix, and consider the problem

$$\min_{U,V} \kappa_2(A + UV^T), \tag{1.1}$$

where $\kappa_2$ denotes the spectral condition number, and $U, V \in \mathbb{R}^{n\times k}$, so that $UV^T$ is a rank-$k$ matrix. The parameter $k$ is prescribed along with the matrix $A$.

The problem of minimizing the condition number of a matrix has been studied often, but not, as far as we know, very systematically. This is in contrast with the problem of minimizing the spectral norm, or the maximum eigenvalue of a symmetric matrix, over given parameterizations. These are much easier problems because the 2-norm is a convex function on the matrix space, as is the maximum eigenvalue on the space of real symmetric matrices, making them amenable to convex optimization techniques (specifically, semidefinite programming); see [1]. By contrast, the condition number is not convex on matrix space. In the case of symmetric positive definite matrices, it is possible to transform a linearly parameterized condition number optimization problem to a convex optimization problem in the semidefinite programming framework (see [1, p. 203]), but when the matrices are symmetric indefinite or nonsymmetric, the problem is more difficult. See [5, 8] for relatively early work on optimizing preconditioners with specified sparsity patterns via eigenvalue optimization. A thorough eigenvalue analysis of low rank perturbations of symmetric matrices is given in the classic [9] and in other places. A recent paper [2] provides necessary and sufficient conditions on when the singular values of a rectangular matrix can be reassigned, using low rank modifications. Low rank perturbations are used, for example, for stable computation of eigenvalues of symmetric tridiagonal matrices using the divide and conquer method [6].

In this paper we provide an analysis of the problem, discuss uniqueness and existence, and derive results for minimizers in a variety of cases, including symmetric semidefinite, symmetric indefinite, and general nonsymmetric matrices, for rank-1, rank-2, and higher rank modifications.

---

*Received by the editors December 13, 2005; accepted for publication (in revised form) by A. J. Wathen June 2, 2006; published electronically December 21, 2006. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/simax/29-1/64755.html

†Department of Computer Science, The University of British Columbia, Vancouver B.C. V6T 1Z4, Canada (greif@cs.ubc.ca, varah@cs.ubc.ca).

There are many reasons for studying condition number optimization, and the recent availability of new approaches and code for solving nonsmooth optimization problems may allow for a comprehensive experimental and theoretical study. Our own original motivation arose from a search for effective preconditioners for symmetric indefinite systems. Of course, the condition number is only one factor in the convergence of iterative solvers, and in fact low rank perturbations by their nature have a limited effect on the spectrum due to the interlacing property which we discuss in detail throughout the paper. Nevertheless, when the spectrum of a matrix does not have an obvious structure, it may be useful to consider whether an approach of condition number minimization is effective, at least for symmetric, or nonsymmetric but normal, matrices.

The rest of the paper is structured as follows. In sections 2–5 we discuss the symmetric problem. First, in section 2 we introduce the interlacing property for rank-$k$ modifications and show how it can be proved using the Courant–Fischer min/max representation. In section 3 we present our analytic results for semidefinite matrices and show that a solution (not necessarily unique) can be obtained by using the eigenvectors corresponding to the smallest eigenvalues. In section 4 we extend our analysis to the symmetric indefinite case and show that a solution can be obtained using the eigenvectors that correspond to the largest and smallest eigenvalues in magnitude. In section 5 we show that even if those eigenvectors are not known exactly, their approximations may yield a nearly optimal solution. In section 6 we show that, using similar techniques, we can deal with the nonsymmetric problem as well. In section 7 we give an example of an application: preconditioning a saddle point system using condition number minimization. We conclude with a short summary of our main observations.

## 2. The interlacing property.

This section and sections 3–5 are devoted to the symmetric version of problem (1.1):

$$\min_{V} \kappa_2(A + VV^T), \tag{2.1}$$

with $A$ symmetric. We will assume throughout that the spectral decomposition of $A$ is given by

$$Q^T A Q = D,$$

with the columns of $Q$ containing the eigenvectors of $A$:

$$Q = [q^{(1)} \quad q^{(2)} \quad \dots \quad q^{(n)}], \qquad i = 1, \dots, n.$$

First consider the rank-1 case. We can write the modified matrix as $(A + \gamma vv^T)$ with $\|v\|_2 = 1$. The following separation theorem, or interlacing property, is well known (see, e.g., [3, p. 442]). Below we provide a proof based on the Courant–Fischer result.

Theorem 2.1. [illegible] $A$ [illegible] $\lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1$ [illegible] $(A + \gamma vv^T)$ [illegible] $\mu_n \le \mu_{n-1} \le \cdots \le \mu_1$, [illegible] $\gamma \ge 0$

$$\lambda_n \le \mu_n \le \lambda_{n-1} \le \mu_{n-1} \le \cdots \le \lambda_1 \le \mu_1.$$

[illegible] For $\gamma \ge 0$, clearly $\mu_i \ge \lambda_i$ for $1 \le i \le n$. To show $\mu_n \le \lambda_{n-1}$, use the Courant–Fischer min/max representation (see [3, p. 394] or [9, p. 101])

$$\lambda_{n-1} = \max_{y \ne 0} \min_{\substack{x^T x = 1 \\ x^T y = 0}} (x^T A x).$$

We have

$$\begin{aligned}\mu_n = \min_{x^Tx=1} x^T(A+\gamma vv^T)x &\le \min_{\substack{x^Tx=1\\ x^Tv=0}} x^T(A+\gamma vv^T)x \\ = \min_{\substack{x^Tx=1\\ x^Tv=0}} (x^TAx) &\le \max_{y\neq 0} \min_{\substack{x^Tx=1\\ x^Ty=0}} (x^TAx) = \lambda_{n-1}.\end{aligned}$$

A similar argument works for the other eigenvalues, using $Y$ with $k$ columns and

$$\lambda_{n-k} = \max_{Y\neq 0} \min_{\substack{x^Tx=1\\ x^TY=0}} (x^TAx).$$

This completes the proof. $\square$

A similar result holds for $\gamma \le 0$. Next, the rank-$k$ case can be handled as a succession of rank-1 modifications:

$$A + VV^T = A + \sum_{j=1}^{k} v^{(j)}{v^{(j)}}^T,$$

where $\{v^{(j)}\}$ are the columns of $V$. Applying the separation theorem successively gives, for example, $\lambda_n \le \mu_n \le \lambda_{n-k}$.

More generally, when $A$ is indefinite, one would like to treat indefinite rank-$k$ modifications

$$A + VEV^T = A + \sum_{j=1}^{k} e_j v^{(j)}{v^{(j)}}^T,$$

where $E = \text{diag}(\pm 1)$. This can also be handled as a succession of rank-1 modifications. Suppose $A_1 = A + v^{(1)}{v^{(1)}}^T$. Then its eigenvalues $\{\mu_j\}$ satisfy

$$\lambda_n \le \mu_n \le \lambda_{n-1} \le \cdots \le \mu_2 \le \lambda_1 \le \mu_1.$$

Now let $A_2 = A_1 - v^{(2)}{v^{(2)}}^T$. Its eigenvalues $\{\tau_j\}$ satisfy

$$\tau_n \le \mu_n \le \tau_{n-1} \le \cdots \le \tau_2 \le \mu_2 \le \tau_1 \le \mu_1.$$

Hence we have

$$\begin{cases} \tau_n \le \lambda_{n-1}, \\ \lambda_n \le \tau_{n-1} \le \lambda_{n-2} \le \cdots \le \lambda_3 \le \tau_2 \le \lambda_1, \\ \lambda_2 \le \tau_1. \end{cases}$$

Similar inequalities hold in the general rank-$k$ case: if $E$ has $p$ positive and $m$ negative coefficients, then in general the eigenvalues $\{\tau_j\}$ of $(A + VEV^T)$ satisfy

$$\lambda_{j+m} \le \tau_j \le \lambda_{j-p}.$$

The proof is a straightforward extension of the above argument and is omitted for the sake of brevity.

The above results provide natural bounds on the eigenvalues of small rank modifications, which we exploit in the following sections.

## 3. The symmetric positive semidefinite case.

Having introduced interlacing results, we now move on to focus on the problem of minimizing the condition number.

### 3.1. Rank-1 modifications.

Consider the problem of minimizing the spectral condition number of a rank-1 modification of a positive semidefinite matrix $A$:

$$\min_v \kappa_2(A + vv^T) = \min_v \frac{\mu_1(v)}{\mu_n(v)}.$$

By scaling $v$, we can express this alternatively as

$$\min_{\substack{\|v\|_2=1\\ \gamma\geq 0}} \kappa_2(A + \gamma vv^T).$$

The case $\gamma \leq 0$ can be handled analogously.

Theorem 3.1. [illegible] $A$ [illegible] $0 \leq \lambda_n \leq \cdots \leq \lambda_1$ [illegible]

$$\min_{\substack{\|v\|_2=1\\ \gamma\geq 0}} \kappa_2(A + \gamma vv^T) = \frac{\lambda_1}{\lambda_{n-1}}$$

[illegible] $v = q^{(n)}$ [illegible] $\lambda_n$ [illegible] $\gamma$ [illegible] $\lambda_{n-1} - \lambda_n \leq \gamma \leq \lambda_1 - \lambda_n$.

[illegible] We can easily see that $\lambda_1/\lambda_{n-1}$ is a lower bound using the interlacing property: we can do no better than keep $\mu_1 = \lambda_1$ and increase $\mu_n$ to $\lambda_{n-1}$, giving $\kappa_2 \geq \frac{\lambda_1}{\lambda_{n-1}}$.

Since $A$ is symmetric, its eigenvectors $\{q^{(i)}\}$ are orthonormal. Hence $(A+\gamma q^{(n)}{q^{(n)}}^T)$ has eigenvalues $\lambda_n + \gamma, \lambda_{n-1}, \ldots, \lambda_2$ and $\lambda_1$. Thus, as long as $\lambda_{n-1} \leq \lambda_n + \gamma \leq \lambda_1$, the extreme eigenvalues are $\lambda_1$ and $\lambda_{n-1}$, so we have equality. ☐

[illegible] The eigenvalues of the modified matrix $(A+\gamma vv^T)$ are generally denoted in [illegible] form as $\mu_n \leq \mu_{n-1} \leq \cdots \leq \mu_1$. However, when $v = q^{(n)}$ as in the proof above, only one of these eigenvalues differs from the original set $\lambda_n \leq \cdots \leq \lambda_1$. In this case it is convenient to refer to that eigenvalue $\lambda_n + \gamma$ as $\mu_n$ (and hence $\mu_i = \lambda_i$ for $i \neq n$) even though the resulting set $\{\mu_n, \ldots, \mu_1\}$ may not be ordered. The interlacing property holds, of course, but since $\gamma$ is such that $\lambda_n + \gamma$ could be larger than $\lambda_{n-1}$, the $\{\mu_i\}$ would have to be renumbered to be properly ordered. We make use of this slight abuse of notation later in section 4.

Finally, note that $\kappa_2(\gamma) \equiv \kappa_2(A + \gamma vv^T)$ has a "flat spot" at its minimum. Of course, one could also consider $\gamma < 0$, which may reduce the condition number further.

### 3.2. Rank-*k* modifications.

For the rank-$k$ case, we consider $\min_V \kappa_2(A + VV^T)$ over all $n \times k$ matrices $V$. If we again scale the columns of $V$, we can express this as

$$\min_{\|v_j\|_2=1} \kappa_2(A + VEV^T) = \min_{\substack{\|v_j\|_2=1\\ \gamma_j\geq 0}} \kappa_2(A + \gamma_1 v_1 v_1^T + \cdots + \gamma_k v_k v_k^T). \tag{3.1}$$

THEOREM 3.2. [illegible] $A$ [illegible] $0 \le \lambda_n \le \cdots \le \lambda_1$ [illegible] $k$ [illegible]

$$\min_V \kappa_2(A + VEV^T) = \frac{\lambda_1}{\lambda_{n-k}}.$$

[illegible] $V = [q^{(n)} \cdots q^{(n-k+1)}]$ [illegible] $E = \mathrm{diag}(\gamma_1, \ldots, \gamma_k)$ [illegible]

$$\left\{ \begin{array}{l} \lambda_{n-k} - \lambda_n \le \gamma_1 \le \lambda_1 - \lambda_n \ ; \\ \\ \lambda_{n-k} - \lambda_{n-k+1} \le \gamma_k \le \lambda_1 - \lambda_{n-k+1}. \end{array} \right. \tag{3.2}$$

[illegible] Again from the interlacing property, $\lambda_1/\lambda_{n-k}$ is a lower bound. Then for $V$ as above, the eigenvalues of $(A+VEV^T)$ are $\lambda_n+\gamma_1, \ldots, \lambda_{n-k+1}+\gamma_k, \lambda_{n-k}, \ldots, \lambda_1$. Thus we must ensure that each transformed eigenvalue is in the closed interval $[\lambda_{n-k}, \lambda_1]$, which is equivalent to requiring (3.2). □

We remark that in this rank-$k$ case, one might also consider instead of (3.1),

$$\min_{\substack{\|v_j\|_2=1 \\ \gamma \ge 0}} \kappa_2(A + \gamma VV^T),$$

with only one scaling factor $\gamma$. Then the above result again applies, but the range of $\gamma$ is more restrictive. For the transformed eigenvalues to lie in $[\lambda_{n-k}, \lambda_1]$, we need $\lambda_n + \gamma \ge \lambda_{n-k}$ [illegible] $\lambda_{n-k+1} + \gamma \le \lambda_1$, or $\lambda_{n-k} - \lambda_n \le \gamma \le \lambda_1 - \lambda_{n-k+1}$. This range is nonempty only if $\lambda_{n-k+1} - \lambda_n \le \lambda_1 - \lambda_{n-k}$, which will certainly be true for $k$ small enough.

These results show that to optimize the condition number of a rank-$k$ modification, one should choose vectors $\{v_j\}$ close to the eigenvectors of $A$ associated with the smallest eigenvalues. In section 5 we will consider the effect of using approximations to these eigenvectors. Note also that solutions to the minimization problem are not necessarily unique: there is a range of values for which the minimum is obtained.

**4. The symmetric indefinite case.** When $A$ is indefinite, we first consider the rank-1 case:

$$\min_{\substack{\|v\|_2=1 \\ \gamma}} \kappa_2(A + \gamma vv^T). \tag{4.1}$$

Here $\gamma$ may be positive or negative. We will keep the ordering of the eigenvalues the same as before (even though some may now be negative) and make the following definitions and assumptions:

- D1. We will assume throughout that $n > 2$. (The case $n = 2$ is trivial.)
- D2. Denote by $\{\sigma_j\}$ the singular values of $A$, $\sigma_1 \ge \cdots \ge \sigma_n \ge 0$. Of course these are simply the moduli of the eigenvalues.
- D3. Let $|\lambda_m| = \min_j |\lambda_j|$. That is, $m$ denotes the index of the smallest eigenvalue in magnitude. Thus $\sigma_n = |\lambda_m|$.
- D4. Assume without loss of generality that the largest eigenvalue in magnitude is $\lambda_n$, so that $\sigma_n = -\lambda_n \ge \lambda_1$. (If not, we can use $-A$ in place of $A$.)
- D5. Finally, since $\kappa_2(A) = \frac{\sigma_1}{\sigma_n} = \frac{|\lambda_n|}{|\lambda_m|}$, we call $\lambda_n$ and $\lambda_m$ the [illegible] eigenvalues of $A$.

The following lemma presents a lower bound for the condition number. Later (in Theorem 4.3) we will show that this bound can actually be attained.

LEMMA 4.1. [illegible] $A$ [illegible]

$$\min_{\substack{\|v\|_2=1\\ \gamma\geq 0}} \kappa_2(A+\gamma vv^T) \geq \frac{\sigma_2}{\sigma_{n-1}}.$$

[illegible] The mapping $A \to (A+\gamma vv^T)$ transforms the eigenvalues from $\{\lambda_i\}$ to $\{\mu_i\}$. Since $\gamma \geq 0$ we have $\mu_i \geq \lambda_i$ for each $i$. In trying to minimize the resulting condition number, we must take into account the interlacing property. Thus, we can do no better than the following:

(i) transform $\lambda_n$ to $\mu_n = -\sigma_2 = \min(\lambda_{n-1}, -\lambda_1)$;
(ii) transform $\lambda_m$ to $\mu_m = \lambda_{m-1}$ (whether $\lambda_m$ is positive or negative);
(iii) leave the other eigenvalues unchanged, i.e., $\mu_i = \lambda_i, i \neq n, m$.

The resulting matrix will have condition number

$$\frac{\max(\lambda_1, |\lambda_{n-1}|)}{\min(|\lambda_{m+1}|, |\lambda_{m-1}|)} = \frac{\sigma_2}{\sigma_{n-1}}, \tag{4.2}$$

which completes the proof. □

[illegible] The case $\gamma \leq 0$ does not change the result, since a similar argument gives a lower bound of $\frac{\max(|\lambda_n|,\lambda_2)}{\min(|\lambda_{m+1}|,|\lambda_{m-1}|)}$. It is worse (greater) than that above, since by one of our assumptions $-\lambda_n \geq \lambda_1$.

We now wish to show that we can actually attain the lower bound by appropriate choice of $v$ (and $\gamma$). We have $\gamma \geq 0$ and thus we can incorporate it into $v$, so that (4.1) becomes

$$\min_v \kappa_2(A+vv^T). \tag{4.3}$$

First consider $A$ diagonal, $A = \mathrm{diag}(d_i)$, $d_n \leq \cdots \leq 0 \leq \cdots \leq d_1$, with active eigenvalues $d_n$ and $d_m$. Now take a vector $v$ with nonzero components only in positions $n$ and $m$. Denote them by $v_n$ and $v_m$. Then $(A+vv^T)$ is identical to $A$ except for the $2\times 2$ block formed by rows and columns $m$ and $n$. This block is

$$\begin{pmatrix} d_m + v_m^2 & v_n v_m \\ v_n v_m & d_n + v_n^2 \end{pmatrix},$$

and its eigenvalues $\mu$ are the roots of

$$\mu^2 - (d_n + d_m + v_m^2 + v_n^2)\mu + d_n d_m + d_n v_m^2 + d_m v_n^2 = 0. \tag{4.4}$$

So the eigenvalues of $(A+vv^T)$ are $\mu_i = d_i$ if $i \neq n, m$ and the roots of (4.4) if $i = n, m$. Let us denote the latter by $\mu_n$ and $\mu_m$, so that $d_n$ is transformed to $\mu_n$ and $d_m$ to $\mu_m$. Although $\mu_n \geq d_n$ and $\mu_m \geq d_m$, they can otherwise be chosen anywhere without violating the interlacing theorem, as again the $\{\mu_i\}$ here are not necessarily ordered.

Evaluating the quadratic equation (4.4) gives two [illegible] equations for the two unknowns $v_m^2$ and $v_n^2$. Fortunately, this linearity, which does not hold in general for cases where $v$ has more than two nonzero components, allows us to make a few useful analytical observations. The linear equations are

$$\begin{pmatrix} d_n-\mu_n & d_m-\mu_n \\ d_n-\mu_m & d_m-\mu_m \end{pmatrix} \begin{pmatrix} v_m^2 \\ v_n^2 \end{pmatrix} = \begin{pmatrix} (\mu_n-d_n)(d_m-\mu_n) \\ (\mu_m-d_n)(d_m-\mu_m) \end{pmatrix}. \tag{4.5}$$

We denote the linear system (4.5) by

$$Bw = c$$

and note that the solution $w$ must have nonnegative components, which restricts the possible choices for $\mu_n$ and $\mu_m$. The case of a homogeneous linear system is trivial, since it implies that $\mu_n = d_n$ and $\mu_m = d_m$, which means none of the eigenvalues change. We therefore have the following result.

LEMMA 4.2. [illegible] $\mu_n$ [illegible] $\mu_m$ [illegible] $\mu_n \le d_m \le \mu_m$ [illegible] $w$ [illegible] $Bw = c$ [illegible]

[illegible] From (4.5) it is easy to see that $\det(B) = (d_m - d_n)(\mu_m - \mu_n) \neq 0$ if $d_m \neq d_n$ and $\mu_m \neq \mu_n$. It is sufficient to consider $d_n < d_m < d_1$, since nonsharp inequalities can be trivially handled separately. By D1–D5 we have $d_n \le -d_1$. Direct computation gives

$$w = B^{-1}c = \frac{1}{d_m - d_n} \begin{pmatrix} (\mu_m-d_m)(d_m-\mu_n) \\ (d_n-\mu_m)(d_n-\mu_n) \end{pmatrix}.$$

Thus $w_1 \ge 0$ if $\mu_n \le d_m \le \mu_m$. Moreover, $w_2 \ge 0$ if $d_n \le \mu_n$ and $d_n \le \mu_m$, but this is ensured since by interlacing the mapping $A \to (A + \gamma vv^T)$ transforms the $\{d_i\}$ into algebraically equal or larger eigenvalues, $\{\mu_i\}$. □

We can therefore choose $\mu_n$ and $\mu_m$ anywhere, subject to the above stated restriction, and are now ready to show that the lower bound presented in Lemma 4.1 can actually be attained.

THEOREM 4.3. [illegible] $A$ [illegible] D1–D5 [illegible]

$$\min_{\substack{\|v\|_2=1 \\ \gamma}} \kappa_2(A+\gamma vv^T) = \min_v \kappa_2(A+vv^T) = \frac{\sigma_2}{\sigma_{n-1}}.$$

[illegible] Since $A$ is symmetric, it can be diagonalized, and so it is reasonable to start by considering a diagonal $A$ as above. In this case we have $\max_{i\neq n}|d_i| = \sigma_2 = \max(d_1, |d_{n-1}|)$, and $\min_{i\neq m}|d_i| = \sigma_{n-1} = \min(|d_{m-1}|, |d_{m+1}|)$. Thus, by Lemma 4.2 we need only ensure that $\mu_n$ and $\mu_m$ do not become active. We must have $\sigma_{n-1} \le |\mu_n|, |\mu_m| \le \sigma_2$, and $\mu_n \le d_m \le \mu_m$. To ensure both of these, choose $\mu_n$ negative, $-\sigma_2 \le \mu_n \le -\sigma_{n-1}$, and $\mu_m$ positive, $\sigma_{n-1} \le \mu_m \le \sigma_2$. Indeed any such choice will result in $\kappa_2(A + vv^T) = \sigma_2/\sigma_{n-1}$, giving a two-parameter family of solutions.

Now, for nondiagonal $A$, suppose $Q^TAQ = D$, diagonal. Then, defining $u = Q^Tv$ we have

$$\begin{aligned} Q^T(A+vv^T)Q &= D + (Q^Tv)(v^TQ) \\ &= D + uu^T. \end{aligned}$$

So, we first solve for $u$ using the above described procedure, giving $u_n$ and $u_m$. We then form

$$v = Qu = u_m q^{(m)} + u_n q^{(n)}.$$

The similarity transformation does not change the 2-norm, and hence not the condition number. It follows that minimizing $\kappa_2(A + vv^T)$ is equivalent to minimizing $\kappa_2(D + uu^T)$, and the proof is complete. □

There are many reasonable choices for $\mu_n$ and $\mu_m$ so that $\mu_n \le d_m \le \mu_m$ and $\sigma_{n-1} \le |\mu_n|, |\mu_m| \le \sigma_2$. For example, one could choose the median singular value $\sigma_* = \sigma_{n/2}$ or $\sigma_{(n+1)/2}$ and pick

$$\mu_n = -\sigma_*, \;\; \mu_m = \sigma_*.$$

It is also worth mentioning that Theorem 4.3 applies to $A$ positive definite, with the resulting modified matrix indefinite.

For the indefinite rank-$k$ case, that is,

$$A + VEV^T = A + \sum_{j=1}^{k} e_j v^{(j)} {v^{(j)}}^T, \tag{4.6}$$

where $E = \text{diag}(e_j) = \text{diag}(\pm 1)$, we first extend the lower bound of Lemma 4.1 as follows.

LEMMA 4.4. [illegible] $A$ [illegible] $VEV^T$ [illegible] $k$ [illegible]

$$\min_{V,E} \kappa_2(A + VEV^T) \ge \frac{\sigma_{k+1}}{\sigma_{n-k}}.$$

[illegible]. Using (4.6) to express $A + VEV^T$ as a sequence of $k$ rank-1 modifications, we apply Lemma 4.1 at each step. Notice that each step can be positive or negative, and the result "peels off" the top and bottom singular values at each step. □

Now, to show that the bound can again be attained, we choose a particular sequence of rank-1 modifications with appropriate sign.

THEOREM 4.5. [illegible] $A$ [illegible] $VEV^T$ [illegible] $k$ [illegible]

$$\min_{V,E} \kappa_2(A + VEV^T) = \frac{\sigma_{k+1}}{\sigma_{n-k}}.$$

[illegible]. For $A$ diagonal, we apply a sequence of $2 \times 2$ rank-1 modifications as in Theorem 4.3, choosing $e_j = +1$ if the largest eigenvalue at that step is negative (and thus transforming eigenvalues into algebraically equal or larger eigenvalues), and $e_j = -1$ if the largest eigenvalue is positive (and thus the eigenvalues are mapped into algebraically equal or smaller eigenvalues). To ensure that we "peel off" the top and bottom singular values at each step, we need only choose the transformed eigenvalues $\mu_n$ and $\mu_m$ so that $\sigma_{n-k} \le |\mu_n|, |\mu_m| \le \sigma_{k+1}$. For a nondiagonal $A$, we again have to multiply by the eigenvector matrix $Q$. □

[illegible] 4.6. Take $A = \text{diag}(-9, -5, -1, 0, 1, 5, 9)$ and ask for the best rank-1 and rank-2 modifications. For the first step, the active eigenvalues can be taken to be $-9$ and $0$. Lemma 4.2 gives ${v^{(1)}}^T = (2.49, 0, 0, 1.67, 0, 0, 0)$ and $e_1 = 1$. The resulting matrix $A_1 = A + v^{(1)}{v^{(1)}}^T$ has eigenvalues $(-5, -5, -1, 1, 5, 5, 9)$, and $\kappa_2(A_1) = 9$. Notice for this example that in this first step, we could choose $\mu_n$ and $\mu_m$ anywhere in the range $1 = \sigma_{n-1} \le |\mu_n|, |\mu_m| \le \sigma_2 = 9$.

For the second step, we take active eigenvalues 9 and $-1$. We get $e_2 = -1$ and ${v^{(2)}}^T = (0, 0, 1.55, 0, 0, 0, 0, 2.37)$. The resulting matrix $A_2 = A_1 - v^{(2)}{v^{(2)}}^T$ has eigenvalues $(-5, -5, -5, 1, 5, 5, 5)$, and $\kappa_2(A_2) = 5$. Notice that we do not have to rediagonalize $A_1$ since the second set of active eigenvalues is distinct from the first.

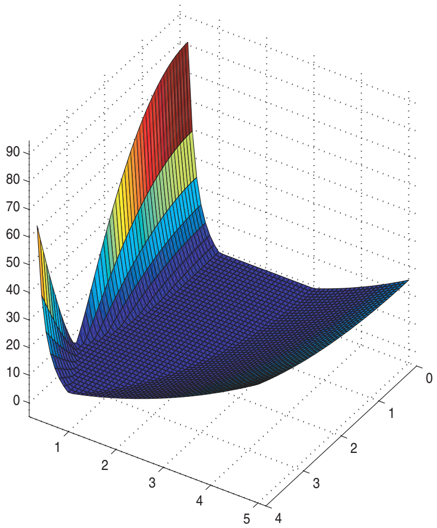


FIG. 4.1. *The condition number of the modified matrix for a range of values of $v_n$ and $v_m$.*

In Figure 4.1 we plot $\kappa_2(A + vv^T)$ for $0 \leq v_m, v_n \leq 4$ . Notice the "flat spot" in this graph, where $\kappa_2 = 9$, that is, a two-dimensional range where the minimum condition is attained. Finally, the solution $V = [v^{(1)} \;\; v^{(2)}]$ is by no means unique. However, it does have the minimum number of nonzero components.

**5. The effect of perturbations.** From the analysis so far, it is clear that to minimize the condition number of the modified matrix, one needs to know particular eigenvectors. Of course, for a large matrix, these are not known explicitly and are often expensive to compute. This raises the question of approximations: what effect will inexact knowledge of the eigenvectors have on the condition number of the modified matrix?

We consider here only the simple case of a semidefinite $A$ (with eigenvalues $\{\lambda_i\}$ and eigenvectors $\{q^{(j)}\}$) modified by a rank-1 matrix. Following Theorem 3.1, we choose the modification $v = q^{(n)}$ and consider

$$C(\gamma) = A + \gamma q^{(n)} {q^{(n)}}^T. \tag{5.1}$$

If $G$ is defined as the interval $\lambda_{n-1} - \lambda_n \leq \gamma \leq \lambda_1 - \lambda_n$, then for $\gamma \in G$, $\lambda_{\min}(C) = \lambda_{n-1}$ and $\lambda_{\max}(C) = \lambda_1$, so $\kappa_2(C)$ is minimized.

Now suppose we allow perturbations in $q^{(n)}$, caused, for example, by inexact approximation. Then we have the following result.

THEOREM 5.1. [illegible] $\gamma \in G$ [illegible]

$$A + \gamma uu^T, \;\; u = q^{(n)} + \varepsilon w,$$

[illegible] $\varepsilon \ll 1$ [illegible] $\|w\|_2 = 1$ [illegible]

$$\kappa_2(A + \gamma uu^T) = \kappa_2(C) + \mathcal{O}(\varepsilon^2), \tag{5.2}$$

[illegible] $C = C(\gamma)$ [illegible] (5.1).

[illegible] Define $F = \gamma(q^{(n)} w^T + w {q^{(n)}}^T)$. Then we have

$$\begin{aligned} A + \gamma u u^T &= A + \gamma(q^{(n)} + \varepsilon w)(q^{(n)} + \varepsilon w)^T \\ &= C(\gamma) + \varepsilon\gamma(q^{(n)} w^T + w {q^{(n)}}^T) + \varepsilon^2 w w^T \\ &= C + \varepsilon F + \varepsilon^2 w w^T. \end{aligned} \tag{5.3}$$

The eigenvalues of $C = C(\gamma)$ are $\lambda_1, \ldots, \lambda_{n-1}$, and $\lambda_n + \gamma$. Now let $\gamma$ be fixed and consider the eigenvalues of the first-order perturbation $C + \varepsilon F$. For any specific eigenvalue $\lambda_j(C)$, let

$$\lambda_j(\varepsilon) = \lambda_j(C + \varepsilon F) = \lambda_j(C) + \varepsilon \lambda_j' + \mathcal{O}(\varepsilon^2).$$

Assuming each eigenvalue is simple, recall that (see, e.g., [9, Chap. 2])

$$\lambda_j' = \frac{{q^{(j)}}^T F q^{(j)}}{{q^{(j)}}^T q^{(j)}}.$$

We have two cases:

(i) $j = n$: $\lambda_n' = \gamma {q^{(n)}}^T (q^{(n)} w^T + w {q^{(n)}}^T) q^{(n)} = 2\gamma w^T q^{(n)}$.

(ii) $j \neq n$: $\lambda_j' = \gamma {q^{(j)}}^T (q^{(n)} w^T + w {q^{(n)}}^T) q^{(j)} = 0$.

Hence

$$\lambda_n(\varepsilon) = \lambda_n(C) + \mathcal{O}(\varepsilon), \quad \lambda_j(\varepsilon) = \lambda_j(C) + \mathcal{O}(\varepsilon^2) \ \text{ for } j \neq n.$$

Thus from (5.3) the same is true for the full perturbation $A + \gamma u u^T$.

Finally then, if $\gamma$ is chosen [illegible] $G$, so that the extreme eigenvalues of $C$ are $\lambda_1$ and $\lambda_{n-1}$, then under perturbation in the vector $u = q^{(n)} + \varepsilon w$, (5.2) follows. □

Theorem 5.1 shows, then, that the effect of a first-order perturbation in the eigenvector is only second-order in the condition number. Thus, an approximation to the eigenvector that can be computed rapidly can be useful for the purpose of obtaining a nearly optimal condition number.

[illegible] 5.2. Consider the discrete Laplace operator using finite difference discretizations on a uniform, two-dimensional grid. It is well known that if Neumann boundary conditions are employed, the matrix has nullity 1 with a vector of constants as its null-space. We set a grid of 32 points in each direction; the resulting matrix is $1024 \times 1024$. The Lanczos algorithm (without reorthogonalization) is applied using four dimension sizes: $k = 4, 8, 16, 32$. The initial guess is random. We compute approximations to the null vector of the matrix using the Ritz vector associated with the smallest Ritz value. As is evident from Table 5.1, the condition number of the modified matrix using the approximation to the null vector is close to that of the modified matrix using the exact null vector, with the relative error decreasing as $\varepsilon$ decreases. A precise assessment of the error is more involved and would require the evaluation of the magnitude of the term multiplied by $\varepsilon^2$ in Theorem 5.1. Nevertheless, for $n$ large enough examining the relative error, $\frac{|\kappa_2(A + q^{(n)} {q^{(n)}}^T) - \kappa_2(A + u u^T)|}{\kappa_2(A + q^{(n)} {q^{(n)}}^T)}$ (given in the last column of the table) illustrates the quadratic dependence on $\varepsilon$, as predicted by Theorem 5.1. For example, between $k = 16$ and $k = 32$ the value of $\varepsilon$ goes down by a factor of approximately 3.43 while the relative error decreases by a factor of approximately 17.5.

Table 5.1

*Effect of perturbations for a discrete Laplace operator with Neumann boundary conditions. The approximations to the null vector are generated using the Lanczos algorithm. In the table, $q^{(n)}$ is a normalized vector of constants (i.e., a null vector of $A$) and $u$ is the approximation to it generated by the Lanczos procedure.*

| $k$ | $\lambda_n$ | $\|Au\|_2$ | $\varepsilon \equiv \|u - q^{(n)}\|_2$ | $\kappa_2(A + q^{(n)}q^{(n)^T})$ | $\kappa_2(A + uu^T)$ | Rel. error |
|---|---|---|---|---|---|---|
| 4 | 3.541e-003 | 0.099821 | 0.053903 | 11.66 | 11.72 | 5.1e-003 |
| 8 | 3.482e-004 | 0.018074 | 0.039415 | 50.55 | 50.63 | 1.6e-003 |
| 16 | 3.637e-005 | 0.0043127 | 0.025597 | 206.17 | 206.30 | 6.3e-004 |
| 32 | 1.629e-006 | 0.00080484 | 0.007465 | 828.69 | 828.72 | 3.6e-005 |

## 6. Extension to nonsymmetric matrices.

We now move to consider the nonsymmetric case. For rank-1 modifications, the nonsymmetric case could be transformed into a problem of minimizing the condition number of a [illegible]-2 modification of the $2n \times 2n$ [illegible] matrix

$$G = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}.$$

But in fact much can be said about the nonsymmetric problem by working on it directly. Consider a nonsymmetric matrix $A$, with singular values $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, and its unsymmetric rank-1 modification $A + uv^T$, with singular values $\tau_1 \geq \cdots \geq \tau_n \geq 0$. Using the interlacing result for symmetric matrices given in section 2, one can formulate a separation theorem for these singular values as well.

Theorem 6.1. [illegible] $\{\tau_j\}$ [illegible] $A + uv^T$ [illegible] $\{\sigma_j\}$ [illegible] $A$ [illegible]

$$\begin{cases} \sigma_2 \leq \tau_1, & \\ \sigma_{k+1} \leq \tau_k \leq \sigma_k, & 1 < k < n, \\ 0 \leq \tau_n \leq \sigma_{n-1}. & \end{cases} \tag{6.1}$$

[illegible] $\kappa_2(A + uv^T)$ [illegible]

$$\kappa_2(A + uv^T) = \frac{\tau_1}{\tau_n} \geq \frac{\sigma_2}{\sigma_{n-1}}. \tag{6.2}$$

[illegible] Using the Courant–Fischer min/max result for $A^TA$,

$$\sigma_{n-k}^2 = \max_{Y \neq 0} \min_{\substack{x^Tx=1 \\ x^TY=0}} (x^TA^TAx)$$

for $Y$ an $n \times k$ matrix. Consider, for example, $\tau_{n-1}$:

$$\begin{aligned} \tau_{n-1}^2 &= \max_{y \neq 0} \min_{\substack{x^Tx=1 \\ x^Ty=0}} x^T(A + uv^T)^T(A + uv^T)x \\ &= \max_{y \neq 0} \min_{\substack{x^Tx=1 \\ x^Ty=0}} x^T\left(A^TA + A^Tuv^T + vu^TA + v(u^Tu)v^T\right)x. \end{aligned}$$

Thus, taking $y = v$,

$$\tau_{n-1}^2 \geq \min_{\substack{x^Tx=1 \\ x^Tv=0}} (x^TA^TAx) \geq \min_{x^Tx=1} (x^TA^TAx) = \sigma_n^2.$$

Moreover,

$$\tau_{n-1}^2 \le \max_{v,y\neq 0} \min_{\substack{x^Tx=1\\ x^Ty=0\\ x^Tv=0}} (x^TA^TAx) \le \max_{\substack{y\neq 0\\ z\neq 0}} \min_{\substack{x^Tx=1\\ x^Ty=0\\ x^Tz=0}} (x^TA^TAx) = \sigma_{n-2}^2.$$

A similar result holds for each intermediate singular value $\tau_k$, $1 < k < n$. For the extreme values $\tau_1$ and $\tau_n$, one achieves only one-sided inequalities; thus $\tau_1 \ge \sigma_2$ and $0 \le \tau_n \le \sigma_{n-1}$. Using these last two inequalities gives (6.2). $\square$

Now we show that this bound can be attained. For $A = D = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$, we proceed as in section 4: consider $(D + uv^T)$ with $u$ and $v$ having nonzero components only in the first and last places, corresponding to the extreme singular values $\sigma_1, \sigma_n$. Then $(D + uv^T)$ is diagonal, with singular values $\sigma_2, \ldots, \sigma_{n-1}$, except for the $2 \times 2$ block

$$\begin{pmatrix} d_1 + u_1v_1 & u_1v_n \\ u_nv_1 & d_n + u_nv_n \end{pmatrix}.$$

Notice that we want to choose $u$ and $v$ so the singular values of this $2\times 2$ block are well inside the interval $[d_n, d_1]$. Choosing $u = v$ does not work, as a positive solution of the analogue of (4.5) results in singular values outside this interval. Thus we need to make the block nonsymmetric but simple enough that the singular values are readily calculated. One approach is to make the block look like $\begin{pmatrix} a & b \\ -b & a \end{pmatrix}$, whose (double) singular values are $(a^2 + b^2)^{1/2}$. For this to happen, we must have

$$\left\{ \begin{array}{l} u_nv_1 = -u_1v_n, \\ d_1 + u_1v_1 = d_n + u_nv_n. \end{array} \right. \tag{6.3}$$

We have two constraints for four unknowns. One way to proceed is to let $u_1, u_n$ be arbitrary, and then (6.3) gives

$$v_1 = \frac{-u_1}{u_1^2 + u_n^2} \cdot (d_1 - d_n), \quad v_n = \frac{u_n}{u_1^2 + u_n^2} \cdot (d_1 - d_n) \tag{6.4}$$

and

$$a = \frac{u_1^2 d_n + u_n^2 d_1}{u_1^2 + u_n^2}, \quad b = \frac{u_1u_n}{u_1^2 + u_n^2} \cdot (d_1 - d_n). \tag{6.5}$$

Notice that the expression for $a$ is a weighted average of $d_1$ and $d_n$ and thus can be made to equal any value in $[d_n, d_1]$ by appropriate choice of $u_1, u_n$.

THEOREM 6.2. [illegible] $A$ [illegible] $n \times n$ [illegible] $(n > 2)$ [illegible] $\sigma_1 \ge \cdots \ge \sigma_n \ge 0$ [illegible]

$$\min_{u,v} \kappa_2(A + uv^T) = \frac{\sigma_2}{\sigma_{n-1}}.$$

[illegible] To show that the bound (6.2) can be attained, use $A$'s singular value decomposition $A = UDV^T$, $D = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$. Then apply the above technique for some value of $\tau^*$, $\sigma_{n-1} \le \tau^* \le \sigma_2$. From (6.5) we have

$$u_1^2 = \frac{d_1^2 - (\tau^*)^2}{d_1^2 - d_n^2}\ , \quad u_n^2 = \frac{(\tau^*)^2 - d_n^2}{d_1^2 - d_n^2}\ , \tag{6.6}$$

and $v_1, v_n$ are given by (6.4). Notice that $u_1^2 + u_n^2 = 1$.

This gives $(D + uv^T)$ with singular values $\sigma_2, \ldots, \sigma_{n-1}$, and $\tau^*$ (twice). Finally,

$$U(D + uv^T)V^T = A + (Uu)(Vv)^T = A + \tilde{u}\tilde{v}^T$$

has minimal condition. Since the 2-norm is invariant under orthogonal transformations, the condition numbers of $D + uv^T$ and $A + \tilde{u}\tilde{v}^T$ are minimized at the same time. $\square$

Notice that $\tilde{u}$ and $\tilde{v}$ are linear combinations of the extreme singular vectors of $A$. In our code we use $\tau^* = \sigma_{n/2}$ or $\sigma_{(n+1)/2}$, the median singular value. Again the solution is *not* unique. The rank-$k$ case can be handled as a sequence of rank-1 modifications, as in previous sections.

*Example* 6.3. The $5 \times 5$ matrix

$$A = \begin{pmatrix} -0.1693 & 0.9417 & -0.5721 & -0.1761 & 0.3667 \\ -0.3900 & 0.9802 & 0.2870 & 0.4891 & -0.5749 \\ 0.7487 & 0.5777 & -0.3599 & -0.4641 & 0.6785 \\ -0.9700 & -0.1227 & 0.9202 & -0.1202 & 0.2576 \\ 0.5359 & -0.0034 & 0.4533 & 0.8668 & -0.7325 \end{pmatrix}$$

was generated randomly, and we sought to minimize the condition number of a rank-1 modification. The singular values of $A$ are $1.8910, 1.5398, 1.4567, 0.6648, 0.1610$. Using our analytical observations and our strategy for choosing $\tau^*$ to be the median singular value, the resulting modified matrix has singular values $1.5398, 1.4567, 1.4567, 1.4567, 0.6648$. By construction, then, we obtain three equal singular values. The optimal condition number is 2.3163. Next, we use the MATLAB command `fminsearch` to find a solution, and get the same minimal value, now with singular values $1.5446, 1.5446, 1.1185, 0.6648, 0.6647$. Thus, the solution is indeed nonunique.

**7. Example: Saddle point system preconditioning.** Consider the numerical solution of a large and sparse saddle point linear system whose associated matrix is

$$\mathcal{K} = \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix},$$

where $A$ is $n \times n$ and $B$ is $n \times m$, with $m < n$. Popular preconditioners have a $2 \times 2$ block diagonal structure, with their (1,1) block approximating the (1,1) block of the original saddle point matrix, and their (2,2) block approximating the Schur complement. Motivated by this, let us make a connection to the analysis presented in the previous sections by considering the preconditioner

$$\mathcal{M} = \begin{pmatrix} A + VV^T & 0 \\ 0 & \pm B^T(A + VV^T)^{-1}B \end{pmatrix},$$

where $V$ is $n \times k$. The $\pm$ signs in front of the $(2,2)$ block suggest two options. It makes sense to consider such a preconditioner if solving a system with $A + VV^T$, a rank-$k$ modification of $A$, is significantly easier than solving for $A$. (Notice that $A$ could be singular even if $\mathcal{K}$ is not.) Thus, we could aim to select a rank-$k$ matrix $V$ that minimizes the condition number of $A + VV^T$.

This approach is computationally delicate for the following reasons. First, $V$ is dense in general, whereas the original saddle point matrix is assumed sparse. In terms of storage, if we are to store $V$ explicitly it will require $nk$ entries. Note that $A + VV^T$

need not be stored explicitly when iterative solvers are used. If $A$ has $\ell$ nonzero entries per row on average, then the storage requirements for the $(1,1)$ block increase from $n\ell$ for $A$ to $n(\ell+k)$ for $A+VV^T$. In terms of computational cost, since a decisive cost factor in the (implicit) inversion of the (1,1) block are matrix-vector products, the overhead for the cost of multiplying a vector by $A+VV^T$ compared to multiplying by $A$ is the addition of two matrix-vector products with $n \times k$ matrices. In other words, the overhead here is $\mathcal{O}(nk)$ floating point operations per iteration. Another potential difficulty is the computation of $V$, which may be expensive to the extent of dominating the cost of solution of the linear system. Here the observations in section 5 come to our aid, since Theorem 5.1 implies that computing $V$ can be done inexactly (likely at a substantially lower cost), while still obtaining a nearly optimal condition number. Finally, to make this approach more practical, inexact inner iterations for solving $A+VV^T$ could be applied throughout the iteration.

The sign in front of the $(2,2)$ block affects the structure of the preconditioned eigenvalues as follows. If it is a positive sign, then the preconditioner is positive definite. In this case the eigenvalues of the preconditioned matrix are real, and a minimum residual solver employing short recurrence relations (such as MINRES) can be applied. If, on the other hand, the sign in front of the $(1,1)$ block is negative, then the preconditioner is no longer positive definite but its inertia is closer to the inertia of the original saddle point matrix. Furthermore, it can be shown that at least $m+n-k$ of the eigenvalues of the preconditioned matrix are complex with unit norm.

Let $\nu$ be an eigenvalue of the preconditioned matrix $\mathcal{M}^{-1}\mathcal{K}$, with associated eigenvector $(x,y)$, and denote

$$M = A + VV^T.$$

Then

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \nu \begin{pmatrix} M & 0 \\ 0 & \pm B^T M^{-1} B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Since we are assuming that $\mathcal{M}^{-1}\mathcal{K}$ is nonsingular, we must have $\nu \neq 0$. Observing that

$$\begin{aligned} &\begin{pmatrix} M & 0 \\ 0 & \pm B^T M^{-1} B \end{pmatrix}^{-1} \begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \\ &\quad = \begin{pmatrix} M & 0 \\ 0 & \pm B^T M^{-1} B \end{pmatrix}^{-1} \left[ \begin{pmatrix} M & B \\ B^T & 0 \end{pmatrix} - \begin{pmatrix} VV^T & 0 \\ 0 & 0 \end{pmatrix} \right], \end{aligned}$$

we now proceed as follows. If the positive sign in front of the $(2,2)$ block of $\mathcal{M}$ is selected, it follows that the preconditioned matrix is a rank-$k$ modification of a matrix which by [7] has precisely three distinct nonzero eigenvalues: 1 and $(1 \pm \sqrt{5})/2$, with algebraic multiplicities $n-m$, $m$, and $m$, respectively. Thus, for $k < \min(m, n-m)$, $\mathcal{M}^{-1}\mathcal{K}$ has eigenvalues 1, $(1 \pm \sqrt{5})/2$ of algebraic multiplicities at least $n-m-k$, $m-k$, and $m-k$, respectively. If, on the other hand, the negative sign is chosen, the preconditioned matrix is a rank-$k$ modification of a matrix with eigenvalues 1 and $\frac{1 \pm \imath\sqrt{3}}{2}$, with the same algebraic multiplicities as above. Here $\imath = \sqrt{-1}$.

Substituting $y = \frac{1}{\nu}(B^T M^{-1} B)^{-1} B^T x$ and defining $\tilde{x} = M^{1/2}x$, we have $(\nu^2 I - \nu K - P)\tilde{x} = 0$, where $K = M^{-1/2} A M^{-1/2}$, $P = P^2 = M^{-1/2} B (B^T M^{-1} B)^{-1} B^T M^{-1/2}$ is an orthogonal projector. In our case

$$K = M^{-1/2} A M^{-1/2} = M^{-1/2}(M - VV^T) M^{-1/2} = I - M^{-1/2} V V^T M^{-1/2} = I - \tilde{V}\tilde{V}^T,$$

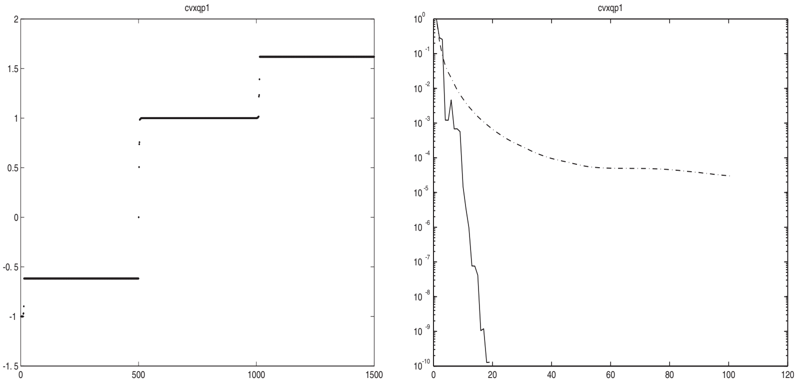


FIG. 7.1. *Eigenvalues of the preconditioned matrix for* `cvxqp1` *(on the left) and convergence of preconditioned MINRES (on the right). The right-hand side vector $b$ was generated by setting the solution as a vector of constants, such that $\|b\|_2 = 1$.*

where $\tilde{V} = M^{-1/2}V$, and we can rewrite our quadratic eigenvalue problem as

$$((\nu^2 - \nu)I + \nu\tilde{V}\tilde{V}^T \mp P)\tilde{x} = 0. \tag{7.1}$$

We can say more if $A$ is symmetric positive semidefinite with nullity $k$. Let $V$ be an $n \times k$ orthogonal matrix representing the null-space of $A$. Since $MV = (A + VV^T)V = V$, it follows that the columns of $V$ are eigenvectors of $M$ with multiple eigenvalues 1. By the analysis of section 3, $V$ is a minimizer for problem (2.1). Since $MV = V$ we have $M^{1/2}V = V$, and hence $\tilde{V} = V$. Thus, (7.1) takes the form

$$((\nu^2 - \nu)I + \nu VV^T \mp P)\tilde{x} = 0.$$

We can thus express the eigenvalue problem in terms of an orthogonal projector onto a space related to the range of $B$ and the null vectors of $A$.

[illegible] 7.1. We used the `cvxqp1` matrix from the CUTEr test collection [4] in its "raw" form, i.e., without taking into account the constraint settings in the context of an optimization problem, for testing the preconditioning approach suggested in this section. The matrix has a $1000 \times 1000$ (1,1) block, whose rank is 986. The size of $B$ is $1000 \times 500$. For this experiment, the matrix $V$ contains the 14 eigenvectors corresponding to the zero eigenvalues. We have applied the preconditioner with a positive sign selected for its $(2, 2)$ block. The eigenvalues of the preconditioned matrix are given in Figure 7.1 on the left and validate the eigenvalue analysis of this section and the algebraic multiplicities of the three clusters of eigenvalues. Convergence graphs for MINRES are given in Figure 7.1 on the right.

Computing the null vectors exactly in this case would be costly and storing all of them would require more storage than that required for the matrix of the linear system. In practice adjustments such as inexpensive approximation of the null vectors and inexact inversion of $A + VV^T$ have to be made. Nevertheless, the substantial savings in iteration counts may indicate the viability of this approach.

**8. Conclusions.** We have considered the problem of minimizing the condition number of a matrix that is subject to low rank modifications. For symmetric matrices, the standard interlacing property of eigenvalues can be applied and for the nonsymmetric case an analogous property of the singular values can be used. There is nonuniqueness, but a solution can be obtained using [illegible] eigenvectors or singular vectors of the matrices, which correspond to extremal eigenvalues (in the symmetric case) or singular values (nonsymmetric case). There is a large "flat spot" of values that can be used to obtain the minimum. In the symmetric indefinite case the two equations that need to be solved to find a possible minimizer are linear, even though the general setting of the problem is nonlinear. For the nonsymmetric case there are more degrees of freedom, and in fact we have four equations with two unknowns. We exploited this freedom by computing the vectors using a particular shifted skew-symmetric matrix for which the singular values are available analytically.

**Acknowledgments.** We are indebted to Michael Overton for many enjoyable and fruitful discussions of the problem—before, after, and while he visited the University of British Columbia in summer 2005. Michael's insightful comments and suggestions for improving this manuscript are gratefully acknowledged. We are also grateful to two very thorough referees for many constructive comments which helped improve this manuscript.

## REFERENCES

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2003.
[2] D. Chu and M. Chu, *Low rank update of singular values*, Math. Comp., 75 (2006), pp. 1351–1366.
[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[4] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.
[5] A. Greenbaum and G. H. Rodrigue, *Optimal preconditioners for a given sparsity pattern*, BIT, 29 (1990), pp. 610–634.
[6] M. Gu and S. C. Eisenstat, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 172–191.
[7] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput., 21 (2000), pp. 1969–1972.
[8] M. L. Overton, *Large-scale optimization of eigenvalues*, SIAM J. Optim., 2 (1992), pp. 88–120.
[9] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 98–116

# SCHWARZ ITERATIONS FOR SYMMETRIC POSITIVE SEMIDEFINITE PROBLEMS*

REINHARD NABBEN† AND DANIEL B. SZYLD‡

**Abstract.** Convergence properties of additive and multiplicative Schwarz iterations for solving linear systems of equations with a symmetric positive semidefinite matrix are analyzed. The analysis presented applies to matrices whose principal submatrices are nonsingular, i.e., positive definite. These matrices appear in discretizations of some elliptic partial differential equations, e.g., those with Neumann or periodic boundary conditions.

**Key words.** linear systems, additive Schwarz, multiplicative Schwarz, domain decomposition methods, symmetric positive semidefinite systems, singular matrices, comparison theorems, overlap, coarse grid correction

**AMS subject classifications.** 65F10, 65F35, 65M55

**DOI.** 10.1137/050644203

## 1. Introduction.

Domain decomposition methods, including additive and multiplicative Schwarz, are widely used for the numerical solution of partial differential equations; see, e.g., [38], [41], [44]. Advantages of these methods include enhancement of parallelism and a localized treatment. One can find algebraic descriptions of them, e.g., in [14], [20], [47], especially for symmetric positive definite problems.

In this paper, we adopt the algebraic representation of additive and multiplicative Schwarz developed in a series of papers [1], [18], [19], [34], [35], where analysis of convergence and properties for several variants of the methods are provided, both for symmetric positive definite and for nonsingular $M$-matrices. Recently, convergence properties were studied for singular systems arising in the solution of Markov chains, i.e., singular $M$-matrices with all principal submatrices being nonsingular [7], [32]. In particular, this theory applies to singular matrices with a one-dimensional null-space, and to those representing irreducible Markov chains; see, e.g., [42]. We also mention the recent work on multiplicative Schwarz iterations for positive semidefinite operators [26], [28].

In this paper, we extend the theory to the symmetric positive semidefinite case, with particular emphasis on the singular case (the analysis of the symmetric positive definite case is known; see, e.g., [1], [21, Ch. 11], [41], [44]). We study in particular the case when all principal submatrices are nonsingular, i.e., positive definite. This situation arises in practice, e.g., in the discretization of certain elliptic differential equations such as $-\Delta u + u = f$ with Neumann or periodic boundary conditions; see, e.g., [5]. We show that in this case, the additive and multiplicative Schwarz iterations are convergent and we characterize the convergence factor $\gamma$ for such methods (sections 4 and 5). We use the theory of matrix splittings (see section 3) to obtain these convergence properties. We remark that we do not use splittings to produce new

---

[*] Received by the editors November 3, 2005; accepted for publication (in revised form) by M. Benzi June 9, 2006; published electronically December 21, 2006.
http://www.siam.org/journals/simax/29-1/64420.html

[†] Institute für Mathematik, Technische Universität Berlin, D-10623 Berlin, Germany (nabben@math.tu-berlin.de).

[‡] Department of Mathematics, Temple University (038-16), 1805 N. Broad St., Philadelphia, PA 19122-6094 (szyld@temple.edu). Supported in part by the U.S. National Science Foundation under grant DMS-0207525, and by the U.S. Department of Energy under grant DE-FG02-05ER25672.

stationary iterative methods. What we do is recast the Schwarz iteration matrices as coming from specific splittings, and we use this setup as an analytical tool to obtain convergence results.

The convergence theory we develop implies that the corresponding preconditioned matrices have zero as an isolated point in the spectrum. The rest of the spectrum is contained in a circle centered at one with radius $\gamma < 1$. When considering additive and multiplicative Schwarz preconditioners for singular systems, one needs to use Krylov subspace methods which are sometimes tailored for this case; see, e.g., [17], [23], [39], and the references given therein.

We believe that our purely algebraic approach is much simpler than that of [26], [28], and in addition, it can be applied to problems which may not have a variational formulation. Of course our approach is only valid for the finite dimensional case. We also consider the case of inexact local solvers (section 6), and the influence of the amount of overlap and the number of blocks in the convergence rate (sections 7 and 8). Finally, we study the convergence of two-level methods, i.e., methods where a coarse grid correction is considered as well (section 9).

**2. The algebraic representation and notation.** We first briefly describe the additive and multiplicative Schwarz methods and give some auxiliary results. Additional notation and background are also given in the next section.

Let $\mathcal{R}(A)$ be the range of $A$. Consider the linear system in $\mathbb{R}^n$ of the form

$$Ax = b, \quad b \in \mathcal{R}(A). \tag{2.1}$$

In this paper we consider the case where $A$ is symmetric positive semidefinite, and we denote this by $A \succeq O$. We assume that every principal submatrix of $A$ is nonsingular, i.e., a symmetric positive definite matrix, and if $A_i$ is such a submatrix, we denote this by $A_i \succ O$. This situation occurs, for instance, when the null-space of $A$, $\mathcal{N}(A)$, is unidimensional and any generator of it has no zero entries; cf. [5].

We consider $p$ subspaces $V_i$, with $\dim V_i = n_i, i = 1, \dots, p$, which are spanned by columns of the identity $I$ over $\mathbb{R}^n$ and such that

$$\sum_{i=1}^{n} V_i = \mathbb{R}^n =: V. \tag{2.2}$$

Note that the subspaces $V_i$ may overlap. Between the subspaces $V_i$ and the space $V$ we consider the following mappings:

$$R_i : V \to V_i, \qquad R_i^T : V_i \to V,$$

where $\operatorname{rank}(R_i^T) = n_i$. $R_i$ is called the restriction operator while $R_i^T$ is called the prolongation operator. We also use the matrices

$$P_i = R_i^T A_i^{-1} R_i A = R_i^T (R_i A R_i^T)^{-1} R_i A,$$

where $A_i := R_i A R_i^T$ is a permutation of a principal submatrix of $A$, which because of our assumption is nonsingular. Note that $P_i$ is a projection.

With these projections the damped additive Schwarz method used as an iterative method to solve (2.1) can be described as

$$\begin{aligned} x^{k+1} &= x^k + \theta \sum_{i=1}^{p} R_i^T A_i^{-1} R_i (b - Ax^k) \\ &= \left( I - \theta \sum_{i=1}^{p} R_i^T A_i^{-1} R_i A \right) x^k + \left( \theta \sum_{i=1}^{p} R_i^T A_i^{-1} R_i \right) b, \end{aligned} \tag{2.3}$$

where $0 < \theta \leq 1$ is a damping parameter; see [8], [11], [12], [13], [20], [21, Ch. 11], [41], [44]. The iteration matrix is then given by

$$T_{AS,\theta} = I - \theta \sum_{i=1}^{p} R_i^T A_i^{-1} R_i A = I - \theta \sum_{i=1}^{p} P_i, \tag{2.4}$$

or, using the notation

$$M_{AS}^{-1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i, \tag{2.5}$$

then, the iteration matrix (2.4) can be written as

$$T_{AS,\theta} = I - \theta M_{AS}^{-1} A.$$

Later on, in Theorem 4.2, we show that the matrix on the right-hand side in (2.5) is nonsingular, and therefore it makes sense to denote it as $M_{AS}^{-1}$. Furthermore, for each $\theta > 0$ one can define a splitting of $A$ for which the iteration matrix is precisely (2.4). One such splitting is $A = \frac{1}{\theta} M_{AS} - (\frac{1}{\theta} M_{AS} - A)$. When $A$ is singular, such splitting however is not unique; see [2].

Very often in practice the additive Schwarz method is used for preconditioning a Krylov subspace method. In the symmetric cases considered here the method of choice is the conjugate gradient method; for a study of this method for singular systems, see [23]. While the matrix $A$ may be singular, the preconditioning matrix $M$ is usually assumed to be symmetric positive definite. The additive Schwarz preconditioner is $M_{AS}^{-1}$ and the preconditioned matrix is then

$$M_{AS}^{-1} A = \sum_{i=1}^{p} P_i = I - T_{AS,1}.$$

The multiplicative Schwarz method can be written as the iteration

$$x^{k+1} = T_{MS} x^k + c, \qquad k = 0, 1, \ldots, \tag{2.6}$$

with the iteration matrix

$$T_{MS} = (I - P_p)(I - P_{p-1}) \cdots (I - P_1) = \prod_{i=p}^{1} (I - P_i), \tag{2.7}$$

and a certain vector $c$. The corresponding preconditioned matrix in this case is $I - T_{MS}$.

[illegible] 2.1. Observe that for any vector $y \in \mathcal{N}(A)$, i.e., such that $Ay = 0$, one has $Ty = y$ for both iteration matrices $T = T_{AS,\theta}$ of (2.4), or $T = T_{MS}$ of (2.7). This implies in particular that we need to require in our iterations, such as (2.3), that $x_0 \notin \mathcal{N}(A)$.

We outline our strategy to prove the convergence of the iterations (2.3) and (2.6). We need to show that the powers of the iteration matrices (2.4) and (2.7) converge to a limit; see Definition 3.1 below. One sufficient condition for this to hold is that there is a splitting of $A$ of the form $A = M - N$ with $M$ nonsingular such that $M^{-1}N$ is the iteration matrix, and we show that this splitting is $P$-regular (see

Definition 3.3 below), which implies convergence; see Theorem 3.2 below. We also use certain comparison theorems to relate the convergence of different versions of these iterations. We present a context for these analytical tools in section 3. In the rest of this section, we repeat the algebraic characterization of the Schwarz methods used, e.g., in [1], which is the basis to produce the above-mentioned splittings.

As already mentioned, we assume that the rows of $R_i$ are rows of the $n \times n$ identity matrix $I$, e.g., of the form

$$R_i = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This restriction operator is often called a Boolean gather operator, while its transpose $R_i^T$ is called a Boolean scatter operator. Formally, such a matrix $R_i$ can be expressed as

$$R_i = [I_i | O]\, \pi_i \tag{2.8}$$

with $I_i$ the identity on $\mathbb{R}^{n_i}$ and $\pi_i$ a permutation matrix on $\mathbb{R}^n$. Then $A_i$ is a symmetric permutation of an $n_i \times n_i$ principal submatrix of $A$. In fact, we can write

$$\pi_i A \pi_i^T = \begin{bmatrix} A_i & K_i \\ K_i^T & A_{\neg i} \end{bmatrix}, \tag{2.9}$$

where $A_{\neg i}$ is the principal submatrix of $A$ "complementary" to $A_i$, i.e.,

$$A_{\neg i} = [O | I_{\neg i}] \cdot \pi_i \cdot A \cdot \pi_i^T \cdot [O | I_{\neg i}]^T$$

with $I_{\neg i}$ the identity on $\mathbb{R}^{n-n_i}$.

For each $i = 1, \ldots, p$, we define

$$E_i := R_i^T R_i \in \mathbb{R}^{n \times n}. \tag{2.10}$$

These diagonal matrices have ones on the diagonal in every row where $R_i^T$ has nonzeros. We further need sets $S_i$ defined by

$$S_i := \{ j \in \{1, \ldots, n\} : (E_i)_{j,j} = 1 \}.$$

Then

$$\bigcup_{i=1}^{p} S_i = S = \{1, 2, \ldots, n\}; \tag{2.11}$$

i.e., each index is in at least one set $S_i$. This is equivalent to saying that $\sum_{i=1}^{p} E_i \geq I$, with equality if and only if there is no overlap. In other words, in the case of overlapping subspaces, we have here that each diagonal entry of $\sum_{i=1}^{p} E_i$ is greater than or equal to one, which implies nonsingularity. Only in the rows corresponding to overlap this matrix has an entry different from one.

For each $i = 1, \ldots, p$, we construct a second set of matrices $M_i \in \mathbb{R}^{n \times n}$ associated with $R_i$ from (2.8) as

$$M_i = \pi_i^T \begin{bmatrix} A_i & O \\ O & D_{\neg i} \end{bmatrix} \pi_i, \tag{2.12}$$

where under our assumptions on $A \succeq O$, we have that $D_{\neg i} = \text{diag}(A_{\neg i}) \succ O$, and thus $M_i$ is invertible.

With the definitions (2.10) and (2.12) we obtain the following equality which we will use throughout the paper:

$$E_i M_i^{-1} A = R_i^T A_i^{-1} R_i A = P_i, \quad i = 1, \ldots, p. \tag{2.13}$$

**3. Convergent matrices, splittings, and comparison theorems.** In this section we present some more definitions and results which we use in the rest of the paper.

Definition 3.1. [illegible] $T$ [illegible] $\lim_{k\to\infty} T^k$ [illegible]

(1) $\rho(T) \le 1$
(2) $\text{rank}(I-T) = \text{rank}(I-T)^2$
(3) [illegible] $|\lambda| = 1$ [illegible] $\lambda$ [illegible] $T$ [illegible] $\lambda = 1$

Condition 2 states that the index of the matrix $I - T$ is one, or in this case that $ind_1 T = 1$ [3]. Several equivalent conditions can be found in [43]. One of them is the following:

$$ind_1 T = 1 \Leftrightarrow \mathcal{R}(I-T) \cap \mathcal{N}(I-T) = \{0\}, \tag{3.1}$$

i.e., that the intersection of the range and the null-space of $I - T$ is trivial.

If $\rho(T) = 1$ for a convergent matrix then the asymptotic rate of convergence is given by

$$\gamma(T) := \max\{|\lambda| \ : \ \lambda \in \sigma(T), |\lambda| < 1\}. \tag{3.2}$$

When $A$ is singular, and we have a nonsingular matrix $M$, and a convergent matrix $T$ such that $A = M(I-T)$, then $P = \lim_{k\to\infty} T^k$ is a projection onto $\mathcal{N}(A) = \mathcal{N}(I-T)$. In fact $P = I-(I-T)(I-T)^D$, where $(I-T)^D$ denote the Drazin inverse of $(I-T)$. Furthermore, if we let $c = M^{-1}b$, and consider the iteration $x^{k+1} = Tx^k + c$, $x_0 \notin \mathcal{N}(A)$ (cf. (2.3)), then $\lim_{k\to\infty} x^k = (I-T)^D c + (I-P)x^0$; see, e.g., [3, Ch. 7.6].

A useful result in the analysis of convergent iteration matrices is the following, due to Keller [24].

Theorem 3.2. [illegible] $A$ [illegible] $M$ [illegible] $M + M^T - A$ [illegible] $T = I - M^{-1}A$ [illegible] $A$ [illegible]

Note than when $M$ is symmetric this theorem says that if $2M - A \succ O$, then $T$ is convergent if and only if $A \succeq O$.

Definition 3.3. [illegible] $A = M - N$ [illegible] $P$-regular [illegible] $M + M^T - A \succ O$ [36] [illegible] strong $P$-regular [illegible] $N \succ O$ [33]

With this definition, Theorem 3.2 indicates that a sufficient condition for convergence of $T$ is that $A = M - N$ is a $P$-regular splitting of a positive semidefinite matrix. Weaker sufficient conditions, and also necessary conditions not requiring the nonsingularity of $M$, can be found in the recent paper [27].

The following result is a new sufficient condition for convergence, which we use later in the paper.

Lemma 3.4. [illegible] $A$ [illegible] $A = M - N$ [illegible] $M$ [illegible]

$$A^{\frac{1}{2}} M^{-1} A^{\frac{1}{2}} \prec 2I,$$

[illegible] $T = I - M^{-1}A$ [illegible] $A = M - N$ [illegible] $P$ [illegible]

We have $A^{\frac{1}{2}}M^{-1}A^{\frac{1}{2}} \prec 2I$. Thus

$$\sigma(A^{\frac{1}{2}}M^{-1}A^{\frac{1}{2}}) \subset [0,2).$$

Since

$$\sigma(A^{\frac{1}{2}}M^{-1}A^{\frac{1}{2}}) = \sigma(M^{-1}A) = \sigma(AM^{-1}) = \sigma(AM^{-\frac{1}{2}}M^{-\frac{1}{2}}) = \sigma(M^{-\frac{1}{2}}AM^{-\frac{1}{2}}),$$

we have that

$$2I - M^{-\frac{1}{2}}AM^{-\frac{1}{2}} \succ 0.$$

Hence,

$$M^{\frac{1}{2}}(2I - M^{-\frac{1}{2}}AM^{-\frac{1}{2}})M^{\frac{1}{2}} \succ 0$$

and therefore,

$$2M - A \succ 0;$$

i.e., we have a $P$-regular splitting. Using Theorem 3.2 we obtain that $T = I - M^{-1}A$ is convergent. □

The use of $P$-regular splittings as sufficient conditions for convergence of classical stationary iterative methods for symmetric matrices mimics the use of regular or weak regular splittings as sufficient conditions for the convergence of classical stationary iterative methods for monotone matrices; see, e.g., the classic books [3], [37], [45]. In this case, the rate of convergence of the iterative method is given by the spectral radius of the iteration matrix. Thus, the rate of convergence of two iterative methods for monotone matrices can be compared by looking at the corresponding spectral radii. Many comparison theorems using different hypothesis on the splittings have appeared in the literature; see, e.g., [9], [10], [16], [29], [33], [45], [46], and other references therein. When the iteration matrices have spectral radius equal to one, as is usually the case for singular linear systems, the convergence rate is given by (3.2). Comparison theorems for these can be found in [30], [31]. Here we present a new comparison theorem, which we use in our context.

We first present the following result due to Weyl; see [22, Theorem 4.3.7]. Let $M \succeq O$, and denote its eigenvalues by $\lambda_1(M) \ge \lambda_2(M), \ldots, \lambda_n(M) \ge 0$.

PROPOSITION 3.5. $M_1$ $M_2$ $M_1 \succeq M_2$ $\lambda_i(M_1) \ge \lambda_i(M_2)$ $i$

Of course, this proposition is valid when $M$ is positive definite as well.

THEOREM 3.6. $A$ $M_1$ $M_2$ $N_1 := M_1 - A$ $N_2 := M_2 - A$

$$M_1^{-1} \succeq M_2^{-1},$$

$$\lambda_i(M_1^{-1}N_1) \le \lambda_i(M_2^{-1}N_2)$$

$i$ $N_1$ $N_2$

$$\gamma(M_1^{-1}N_1) \le \gamma(M_2^{-1}N_2).$$

[illegible] We first note that

$$\sigma(M_k^{-1}A) = \sigma(M_k^{-1}A^{\frac{1}{2}}A^{\frac{1}{2}}) = \sigma(A^{\frac{1}{2}}M_k^{-1}A^{\frac{1}{2}}), \quad k = 1, 2.$$

With Proposition 3.5 we obtain for each $i$ that

$$\lambda_i(M_1^{-1}A) = \lambda_i(A^{\frac{1}{2}}M_1^{-1}A^{\frac{1}{2}}) \geq \lambda_i(A^{\frac{1}{2}}M_2^{-1}A^{\frac{1}{2}}) = \lambda_i(M_2^{-1}A). \tag{3.3}$$

Since $M_k^{-1}N_k = I - M_k^{-1}A$, $k = 1, 2$, (3.3) indicates that for each $i$,

$$\lambda_i(M_1^{-1}N_1) \leq \lambda_i(M_2^{-1}N_2).$$

If $N_1$ and $N_2$ are positive semidefinite then all eigenvalues of $M_1^{-1}N_1$ and $M_2^{-1}N_2$ are nonnegative, and therefore

$$\gamma(M_1^{-1}N_1) \leq \gamma(M_2^{-1}N_2). \qquad \square$$

## 4. Convergence of additive Schwarz.

We begin with an auxiliary result, the proof of which follows by a straightforward calculation.

LEMMA 4.1. [illegible] $A$ [illegible]

$$A^{\frac{1}{2}}R_i^T(R_iAR_i^T)^{-1}R_iA^{\frac{1}{2}}$$

[illegible] $I - A^{\frac{1}{2}}R_i^T(R_iAR_i^T)^{-1}R_iA^{\frac{1}{2}}$ [illegible]

$$A^{\frac{1}{2}}R_i^T(R_iAR_i^T)^{-1}R_iA^{\frac{1}{2}} \preceq I, \tag{4.1}$$

[illegible]

$$\sigma(A^{\frac{1}{2}}R_i^T(R_iAR_i^T)^{-1}R_iA^{\frac{1}{2}}) = \{0, 1\}.$$

THEOREM 4.2. [illegible] $A$ [illegible] $b \in \mathcal{R}(A)$ [illegible] $x_0 \notin \mathcal{N}(A)$ [illegible] $0 < \theta < 2/p$ [illegible] (2.4) [illegible] $M = \frac{1}{\theta}M_{AS}$ [illegible] $P$ [illegible]

[illegible] First, as is done in [21] for the nonsingular case, we prove that the matrix

$$\sum_{i=1}^{p} R_i^T(R_iAR_i^T)^{-1}R_i$$

is nonsingular. To that end, let the vector $x$ be such that

$$\sum_{i=1}^{p} R_i^T(R_iAR_i^T)^{-1}R_ix = 0.$$

Hence

$$x^T\sum_{i=1}^{p} R_i^T(R_iAR_i^T)^{-1}R_ix = 0,$$

and thus

$$\sum_{i=1}^{p}(A_i^{-\frac{1}{2}}R_ix)^TA_i^{-\frac{1}{2}}R_ix = \sum_{i=1}^{p}||A_i^{-\frac{1}{2}}R_ix||_2^2 = 0,$$

which implies $R_i x = 0$ for $i = 1, \ldots, p$. By our assumption (2.2) this implies that $x = 0$.

Using Lemma 4.1 we have that (4.1) holds. Summing up, we have

$$A^{\frac{1}{2}} \left( \sum_{i=1}^{p} R_i^T (R_i A R_i^T)^{-1} R_i \right) A^{\frac{1}{2}} \preceq pI, \tag{4.2}$$

and since $\theta < 2/p$, we have $A^{\frac{1}{2}} \theta M_{AS}^{-1} A^{\frac{1}{2}} \prec 2I$. We can now use Lemma 3.4, and this completes the proof. □

As is done in [21, Ch. 11.2.4] in the symmetric positive definite case, a careful look at the sum in (4.2) indicates that we can replace the number of subdomains $p$ with the number of colors $q$ of the graph of $A$. Thus $A^{\frac{1}{2}} M_{AS}^{-1} A^{\frac{1}{2}} \prec qI$, and if $\theta < 2/q$, we have convergence.

[illegible] 4.3. If we further restrict the value of the damping parameter to $\theta < 1/p$ (or $\theta < 1/q$), we have that the splitting defined by $\frac{1}{\theta} M_{AS}$ is strong $P$-regular. This follows since in this case $A^{\frac{1}{2}} \theta M_{AS}^{-1} A^{\frac{1}{2}} \prec I$, which implies $\frac{1}{\theta} M_{AS} \succ A$.

We note that the result in Theorem 4.2 applies in particular to the symmetric positive definite case. Thus, in our formulation we have doubled the interval of admissible damping factors for convergence of the damped additive Schwarz method, since the usual restriction is that $\theta < 1/q$; see [18], [21, Ch. 11.2.4]. We mention also that simple examples show that this method may not be convergent for $\theta = 1$.

From Theorem 4.2 it follows that the only eigenvalue of $T$ in the unit circle is $\lambda = 1$, and since we showed that $M_{AS}$ is nonsingular, the corresponding eigenvector is a generator of the one-dimensional $\mathcal{N}(A)$. It follows then (see, e.g., [22, section 4.2]), that the convergence factor (3.2) of the additive Schwarz iteration can be characterized as

$$\begin{aligned} \gamma(T_{AS,\theta}) &= \max_{\substack{z \perp \mathcal{N}(A) \\ z^T z = 1}} z^T T_{AS,\theta} z \\ &= \max_{\substack{z \perp \mathcal{N}(A) \\ (z,z) = 1}} \left( 1 - \theta \sum_{i=1}^{p} (R_i^T A_i^{-1} R_i z, Az) \right) \\ &= 1 - \theta \left( \min_{\substack{z \perp \mathcal{N}(A) \\ (z,z) = 1}} \sum_{i=1}^{p} (R_i^T A_i^{-1} R_i z, Az) \right). \end{aligned} \tag{4.3}$$

We note that on the subspace $\mathcal{N}(A)^{\perp}$, the matrix $A$ is positive definite. Let us call $\hat{A} = A|_{\mathcal{N}(A)^{\perp}}$, and we can thus replace $A$ with $\hat{A}$ in (4.3). Furthermore, since $\hat{A}^{1/2}$ is invertible, we can write $w = \hat{A}^{1/2} z$, and write (4.3) as

$$\gamma(T_{AS,\theta}) = 1 - \theta \left( \min_{\substack{\hat{A}^{-1/2} w \perp \mathcal{N}(A) \\ (w, \hat{A}^{-1} w) = 1}} \sum_{i=1}^{p} w^T \hat{A}^{-1/2} R_i^T A_i^{-1} R_i \hat{A}^{1/2} w \right). \tag{4.4}$$

We point out that the characterization (4.4) is also valid for the case of $A$ symmetric positive definite, in which case we have $\hat{A} = A$.

**5. Convergence of multiplicative Schwarz.** We begin with an important auxiliary result.

LEMMA 5.1. [illegible] $A$ [illegible] $x,\ y \in \mathbb{R}^n$ [illegible]

$$y = (I - E_i M_i^{-1} A)x, \tag{5.1}$$

[illegible] $E_i$ [illegible] (2.10) [illegible] $M_i$ [illegible] (2.12) [illegible]

$$y^TAy - x^TAx = -(y-x)^TE_iAE_i(y-x) \le 0. \tag{5.2}$$

[illegible] Consider $x = \pi_i^T(x_1^T, x_2^T)^T$ and $y = \pi_i^T(y_1^T, y_2^T)^T$, with $x_1,\ y_1 \in \mathbb{R}^{n_i}$. Further, from (2.10) and (2.8) we have that

$$E_i = \pi_i^T \begin{bmatrix} I_i & O \\ O & O \end{bmatrix} \pi_i. \tag{5.3}$$

Consider now (5.1), whence we immediately have that

$$y_2 = x_2, \tag{5.4}$$

and using (2.12) and (2.9), we also get

$$A_iy_1 = -A_{12}x_2, \tag{5.5}$$

where here we use the notation $A_{12} = K_i$, and similarly $A_{21} = K_i^T = A_{12}^T$. Using these identities we write

$$\begin{aligned} y^TAy - x^TAx &= (y_1^T, y_2^T)\pi_iA\pi_i^T(y_1^T, y_2^T)^T - (x_1^T, x_2^T)\pi_iA\pi_i^T(x_1^T, x_2^T)^T \\ &= y_1^TA_iy_1 + y_2^TA_{21}y_1 + y_1^TA_{12}y_2 - x_1^TA_ix_1 - x_2^TA_{21}x_1 - x_1^TA_{12}x_2 \\ &= x_2^TA_{21}(y_1 - x_1) + (y_1^T - x_1^T)A_{12}x_2 + y_1^TA_iy_1 - x_1^TA_ix_1 \\ &= -y_1^TA_i(y_1 - x_1) - (y_1^T - x_1^T)A_iy_1 + y_1^TA_iy_1 - x_1^TA_ix_1 \\ &= -(y_1^T - x_1^T)A_i(y_1 - x_1) = -(y-x)^TE_iAE_i(y-x), \end{aligned}$$

where the last equality follows from the identity

$$E_iAE_i = \pi_i^T \begin{bmatrix} A_i & O \\ O & O \end{bmatrix} \pi_i.$$

Since $A \succeq O$, $E_iAE_i$ is semidefinite as well, and the right-hand side of (5.2) is nonpositive. ◻

THEOREM 5.2. [illegible] $A$ [illegible] $b \in \mathcal{R}(A)$ [illegible] $x_0 \notin \mathcal{N}(A)$ [illegible] (2.6) [illegible]

[illegible] We need to prove that the iteration matrix $T = T_{MS}$ is convergent; i.e., we need to prove conditions (1), (2), and (3) of Definition 3.1.

(1) Starting with $z = x^{(1)} \notin \mathcal{N}(A)$, let $x^{(i+1)} = (I - P_i)x^{(i)}$. Thus $x^{(p+1)} = Tx^{(1)}$. Using (5.2) repeatedly, and canceling terms, we obtain

$$\begin{aligned} z^TT^TATz - z^TAz &= -\sum_{i=1}^{p}(x^{(i+1)} - x^{(i)})^TE_iAE_i(x^{(i+1)} - x^{(i)}) \\ &= -\sum_{i=1}^{p}((x^{(i+1)} - x^{(i)})^TE_i)E_iAE_i(E_i(x^{(i+1)} - x^{(i)})). \end{aligned} \tag{5.6}$$

Since $E_iAE_i$ is positive definite it follows that the right-hand side of (5.6) is nonpositive. However, the right-hand side is zero if and only if

$$E_i(x^{(i+1)} - x^{(i)}) = 0 \qquad \text{for all } i,\ i = 1, \ldots, p.$$

The other $n - n_i$ components of $x^{(i+1)} - x^{(i)}$ are also zero using the same argument as in Lemma 5.1 to obtain (5.4). But this implies $x^{(p+1)} = x^{(i+1)} = x^{(i)} = x^{(1)}$, $i = 1, \ldots, p$. Thus $x^{(1)}$ must be a common fixed point of $(I - P_i)$ for all $i = 1, \ldots, p$. However, the fixed points of the projections $(I - P_i)$ are just the vectors $z \in \mathbb{R}^n$ with $E_i z = 0$. Since $\sum_{i=1}^{p} E_i \geq I$ there is no such common nonzero fixed point. Hence the right-hand side of (5.6) must be negative, and we obtain

$$z^T T^T A T z - z^T A z < 0.$$

Thus we have that for all $\lambda \in \sigma(T)$ with corresponding eigenvector $y \notin \mathcal{N}(A)$

$$\lambda^2 y^T A y - y^T A y < 0. \tag{5.7}$$

Hence $\lambda^2 - 1 < 0$. Thus

$$|\lambda| < 1.$$

If $\lambda \in \sigma(T)$ but the corresponding eigenvector $y \in \mathcal{N}(A)$, we easily obtain from the definition of $T$ that $\lambda = 1$. Hence, $\rho(T) \leq 1$.

(2) By (3.1), it suffices to prove that $\mathcal{N}(I - T) \cap \mathcal{R}(I - T) = \{0\}$. Here we have that $\mathcal{N}(A) = \mathcal{N}(I - T)$. This holds since $y \notin \mathcal{N}(A)$ implies $Ty \neq y$ by part (1), i.e., $y \notin \mathcal{N}(I - T)$. On the other hand $y \in \mathcal{N}(A)$ implies $y \in \mathcal{N}(I - T)$, using the definition of $T$; cf. Remark 2.1. Hence, we need to prove that

$$\mathcal{N}(A) \cap \mathcal{R}(I - T) = \{0\}. \tag{5.8}$$

Let $x \in \mathcal{N}(A) \cap \mathcal{R}(I - T)$. Then there exists a $y$ with $(I - T)y = x$, i.e., $y = Ty + x$. Since $x \in \mathcal{N}(A)$ we obtain

$$A(I - T)y = Ax = 0, \quad \text{and thus} \quad y^T A y - y^T A T y = 0.$$

Using $y = Ty + x$ we get

$$y^T A y - y^T T^T A T y + x^T A T y = y^T A y - y^T T^T A T y = 0.$$

Part (1) of this proof now implies $y \in \mathcal{N}(A)$; cf. (5.7). Therefore, by Remark 2.1, $x = (I - T)y = 0$, which completes this part of the proof.

(3) As proved above we have $\lambda < 1$ for all $\lambda \in \sigma(T)$ with corresponding eigenvector $y \notin \mathcal{N}(A)$. Thus if $|\lambda| = 1$ for some eigenvalue $\lambda$ of $T$ then the corresponding eigenvector $y$ must be in the null-space of $A$. Hence $Ay = 0$. But then $Ty = y$ and thus $\lambda = 1$. ☐

We mention that we need to prove explicitly (5.8) since we do not have an explicit representation of a nonsingular matrix $M_{MS}$ such that $M_{MS}^{-1} A = I - T_{MS}$. The existence of such a matrix, i.e., of a splitting induced by $T_{MS}$ [2] is only obtained after the theorem is proved. Any splitting induced by such a matrix $M_{MS}$ is thus $P$-regular.

We also comment on the fact that in some cases one may want to have a symmetric operator, and in such a case, the natural multiplicative operator is

$$T_{SMS} = (I - P_1)(I - P_2) \cdots (I - P_{p-1})(I - P_p)(I - P_{p-1}) \cdots (I - P_1). \tag{5.9}$$

It follows that Theorem 6.1 applies to this case as well, and that a posteriori, there exists a nonsingular matrix $M_{SMS}$ such that $M_{SMS}^{-1} A = I - T_{SMS}$. We can characterize the convergence factor (3.2) of this symmetric multiplicative Schwarz iteration as

$$\gamma = \gamma(T_{SMS}) = \max_{\substack{z \perp \mathcal{N}(A) \\ z^T z = 1}} (z, T_{SMS} z). \tag{5.10}$$

**6. Inexact local solvers.** In this section we study the effect of varying how exactly (or inexactly) the local problems are solved. The convergence of these very practical versions of the methods is based on the same ideas used to prove that of the standard Schwarz iterations in sections 4 and 5. The influence of different levels of inexactness is analyzed using our comparison theorem, Theorem 3.6.

Very often in practice, instead of solving the local problems $A_i y_i = z_i$ exactly, such linear systems are approximated by $\tilde{A}_i^{-1} z_i$, where $\tilde{A}_i$ is an approximation of $A_i$; see, e.g., [6], [41], [44]. The expression $\tilde{A}_i^{-1} z_i$ often represents an approximation to the solution of the system $A_i z_i = v_i$ using some steps of an (inner) iterative method. By replacing $A_i$ with $\tilde{A}_i$ in (2.4) one obtains the damped additive Schwarz iterations with inexact local solvers, and its iteration matrix is then

$$\tilde{T}_{AS,\theta} = I - \theta \sum_{i=1}^{p} R_i^T \tilde{A}_i^{-1} R_i A. \tag{6.1}$$

The iteration matrices $T_{AS,\theta}$ and $\tilde{T}_{AS,\theta}$ in (2.4) and (6.1) are induced by splittings $A = M_\theta - N_\theta$ and $A = \tilde{M}_\theta - \tilde{N}_\theta$ where

$$M_\theta^{-1} = \theta \sum_{i=1}^{p} R_i^T A_i^{-1} R_i = \theta \sum_{i=1}^{p} E_i M_i^{-1} \succ O, \tag{6.2}$$

$$\tilde{M}_\theta^{-1} = \theta \sum_{i=1}^{p} R_i^T \tilde{A}_i^{-1} R_i = \theta \sum_{i=1}^{p} E_i \tilde{M}_i^{-1} \succ O. \tag{6.3}$$

Here

$$\tilde{M}_i = \pi_i^T \begin{bmatrix} \tilde{A}_i & O \\ O & D_{\neg i} \end{bmatrix} \pi_i, \quad \text{and thus} \quad \tilde{M}_i^{-1} = \pi_i^T \begin{bmatrix} \tilde{A}_i^{-1} & O \\ O & D_{\neg i}^{-1} \end{bmatrix} \pi_i. \tag{6.4}$$

The fact that the matrix (6.3) is nonsingular follows in the same manner as in the proof that (6.2) is nonsingular in Theorem 4.2.

In the case considered in this paper we assume, as is generally done (see, e.g., [21, Ch. 11.2.4]), that the inexact local solvers correspond to symmetric positive definite matrices and satisfy

$$\tilde{A}_i \succeq A_i. \tag{6.5}$$

For examples of splittings for which the inequality (6.5) holds, see, e.g., [33]. A situation worth mentioning where (6.5) holds is when $A_i$ is semidefinite and the inexact local solver is definite. This process is usually called regularization; see, e.g., [15], [25].

Theorem 6.1. [illegible] $A$ [illegible] $b \in \mathcal{R}(A)$ [illegible] $x_0 \notin \mathcal{N}(A)$ [illegible] $\tilde{A}_i$ [illegible] $\bar{A}_i$ [illegible] $A_i$ [illegible] $\tilde{A}_i \succeq \bar{A}_i \succeq A_i$ [illegible] $\bar{T}_{AS,\theta}$ [illegible] $\tilde{A}_i$ [illegible] $\bar{A}_i$ [illegible] (6.1) $i = 1, \ldots, p$ [illegible] $0 < \theta < 2/p$ [illegible] (6.1) [illegible] $\bar{T}_{AS,\theta}$ [illegible] $P$ [illegible] $0 < \theta < 1/p$ [illegible] $\gamma(T_{AS,\theta}) \le \gamma(\bar{T}_{AS,\theta}) \le \gamma(\tilde{T}_{AS,\theta})$ [illegible] $P$ [illegible]

[illegible] Since $\tilde{A}_i \succeq A_i$ we have

$$\tilde{A}_i^{-1} \preceq A_i^{-1}, \tag{6.6}$$

and thus, using Lemma 4.1

$$A^{\frac{1}{2}} R_i^T \tilde{A}_i^{-1} R_i A^{\frac{1}{2}} \preceq A^{\frac{1}{2}} R_i^T A_i^{-1} R_i A^{\frac{1}{2}} \preceq I.$$

Similar inequalities are obtained with $\bar{A}_i$. The rest of the convergence proof proceeds in the same manner as that of Theorem 4.2.

Consider the matrices (6.2) and (6.3) which are symmetric positive definite using $M_i$ as in (2.12) and $\tilde{M}_i$ as in (6.4). From (6.6), we have that $M_\theta^{-1} \succeq \tilde{M}_\theta^{-1} \succ O$. This implies $M_\theta \preceq \tilde{M}_\theta$ and $N_\theta \preceq \tilde{N}_\theta$. By Remark 4.3, we have that $N_\theta \succ O$, i.e., that the splittings are strong $P$-regular. The same results are obtained in the case of $\bar{A}_i$. The theorem follows from Theorem 3.6. $\square$

As was the case with Theorem 4.2, we can replace $p$ in the restriction on the damping parameter with $q$, the number of colors; i.e., we guarantee convergence of additive Schwarz with inexact local solvers for $\theta < 2/q$. Since Theorem 6.1 applies in particular to the symmetric positive definite case, we have again double the interval of admissible damping factors for the additive Schwarz iteration with inexact local solvers; cf. [1].

*Remark* 6.2. An alternative proof of the second part of Theorem 6.1 can be obtained by considering the two convergence factors, $\gamma(T_{AS,\theta})$ given by (4.4) for the exact case, and the second given by

$$\gamma(\tilde{T}_{AS,\theta}) = 1 - \theta \left( \min_{\substack{\hat{A}^{-1/2} w \perp \mathcal{N}(A) \\ (w, \hat{A}^{-1} w) = 1}} \sum_{0=1}^{p} w^T \hat{A}^{-1/2} R_i^T \tilde{A}_i^{-1} R_i \hat{A}^{1/2} w \right) \tag{6.7}$$

for the inexact case. Since $\sigma(\hat{A}^{-1/2} R_i^T A_i^{-1} R_i \hat{A}^{1/2}) = \{0\} \cup \sigma(A_i^{-1})$ and $\sigma(\hat{A}^{-1/2} R_i^T \tilde{A}_i^{-1} R_i \hat{A}^{1/2}) = \{0\} \cup \sigma(\tilde{A}_i^{-1})$, and since $-\tilde{A}_i^{-1} \succeq -A_i^{-1}$, we have that

$$-w^T \hat{A}^{-1/2} R_i^T \tilde{A}_i^{-1} R_i \hat{A}^{1/2} w \ \geq \ - w^T \hat{A}^{-1/2} R_i^T A_i^{-1} R_i \hat{A}^{1/2} w, \quad i = 1, \ldots, p,$$

which implies that $\gamma(\tilde{T}_{AS,\theta}) \geq \gamma(T_{AS,\theta})$.

For simplicity, in Theorem 6.1, we assumed that the inexact versions use the same damping parameter $\theta$. It is evident from the proofs that if the damping parameter for the inexact version is smaller, say, $\tilde{\theta} < \theta$, the same conclusions hold.

The implication of Theorem 6.1 is that by replacing the local solvers $A_i$ with the approximate counterparts $\tilde{A}_i$, the additive Schwarz iteration is expected to take more iterations. In practice, a solve with $\tilde{A}_i$ should be sufficiently less expensive so that the overall method is cheaper.

Next we consider the multiplicative Schwarz method with inexact local solvers on the subdomains. Here we assume that the approximations $\tilde{A}_i$ satisfy

$$\tilde{A}_i + \tilde{A}_i^T - A_i \succ 0. \tag{6.8}$$

This assumption implies that

$$A_i = \tilde{A}_i - (\tilde{A}_i - A_i) \quad \text{are } P\text{-regular splittings.}$$

Using (6.4), the inexact multiplicative Schwarz iteration matrix is given by

$$\tilde{T} = (I - E_p \tilde{M}_p^{-1} A)(I - E_{p-1} \tilde{M}_{p-1}^{-1} A) \cdots (I - E_1 \tilde{M}_1^{-1} A). \tag{6.9}$$

LEMMA 6.3. [illegible] $A$ [illegible] $x,\ y \in \mathbb{R}^n$ [illegible] $y = (I - E_i\tilde{M}_i^{-1}A)x$ [illegible] $\tilde{M}_i$ [illegible] (6.4) [illegible] $\tilde{A}_i$ [illegible] (6.8) [illegible]

$$-(y-x)^T E_i(\tilde{M}_i^T + \tilde{M}_i - A)E_i(y-x) \le 0. \tag{6.10}$$

[illegible] The proof proceeds as that of Lemma 5.1. We have that (5.4) holds, but instead of (5.5) we have $\tilde{A}_i y_1 = (\tilde{A}_i - A_i)x_1 - A_{12}x_2$. We then obtain

$$\begin{aligned} y^TAy - x^TAx &= x_2^TA_{21}(y_1-x_1) + (y_1^T - x_1^T)A_{12}x_2 + y_1^TA_iy_1 - x_1^TA_ix_1 \\ &= (x_1^T(\tilde{A}_i - A_i)^T - y_1^T\tilde{A}_i^T)(y_1-x_1) \\ &\quad + (y_1^T - x_1^T)((\tilde{A}_i - A_i)x_1 - \tilde{A}_iy_1) + y_1^TA_iy_1 - x_1^TA_ix_1 \\ &= (-x_1^TA_i - (y_1^T - x_1^T)\tilde{A}_i^T)(y_1 - x_1) \\ &\quad + (y_1^T - x_1^T)(-A_ix_1 - \tilde{A}_i(y_1 - x_1)) + y_1^TA_iy_1 - x_1^TA_ix_1 \\ &= -(y_1^T - x_1^T)(\tilde{A}_i + \tilde{A}_i^T - A_i)(y_1 - x_1) \\ &= -(y-x)^TE_i(\tilde{M}_i^T + \tilde{M}_i - A)E_i(y-x) \le 0, \end{aligned}$$

where the last inequality follows from (6.8) and the form of the matrices $\tilde{M}_i$ in (6.4). □

THEOREM 6.4. [illegible] $A$ [illegible] $b \in \mathcal{R}(A)$ [illegible] $x_0 \notin \mathcal{N}(A)$ [illegible] (6.9) [illegible] $\tilde{M}_i$ [illegible] (6.4) [illegible] (6.8) [illegible] $Ax = b$

[illegible] We need to prove that the iteration matrix $T$ is convergent; i.e., we need to prove conditions (1), (2), and (3) of Definition 3.1. The proof is similar to the proof of Theorem 5.2. The only difference appears in proving condition (1). Here we use Lemma 6.3 and obtain

$$z^T\tilde{T}^TA\tilde{T}z - z^TAz < 0$$

for all $z \notin \mathcal{N}(A)$, and the rest of the proof follows. □

A symmetric version of multiplicative Schwarz with inexact local solvers can also be constructed in a way similar to (5.9), and its convergence factor can be characterized in a way similar to (5.10).

We mention that a comparison analogous to that of the second part of Theorem 6.1 is not valid for multiplicative Schwarz, not even in the definite case. A counterexample can be found in [40].

## 7. Varying the amount of overlap.

We study here how varying the amount of overlap between subblocks (subdomains) influences the convergence rate of additive Schwarz.

Let us consider two sets of subblocks (subdomains) of the matrix $A$, as defined by the sets (2.11), such that one has more overlap than the other; i.e., let

$$\hat{S}_i \supseteq S_i, \quad i = 1, \ldots, p, \tag{7.1}$$

with $\bigcup_{i=1}^p \hat{S}_i = \bigcup_{i=1}^p S_i = S$. Of course, each set $\hat{S}_i$ defines an $\hat{n}_i \times n$ matrix $\hat{R}_i$, where $\hat{n}_i$ is the cardinality of $\hat{S}_i$, and the corresponding $n \times n$ matrix $\hat{E}_i = \hat{R}_i^T\hat{R}_i$, as in (2.10). The relation (7.1) implies that

$$I \succeq \hat{E}_i \succeq E_i \succeq O. \tag{7.2}$$

Similarly, if $\hat{\pi}_i$ is such that $\hat{R}_i = [I_i|O]\,\hat{\pi}_i$, with $I_i$ the identity in $\mathbb{R}^{\hat{n}_i}$, we denote by $\hat{A}_i$ the corresponding principal submatrix of $A$, i.e.,

$$\hat{A}_i = \hat{R}_i A \hat{R}_i^T = [I_i|O] \cdot \hat{\pi}_i \cdot A \cdot \hat{\pi}_i^T \cdot [I_i|O]^T,$$

and, as in (2.12) define

$$\hat{M}_i = \hat{\pi}_i^T \begin{bmatrix} \hat{A}_i & O \\ O & \hat{D}_{\neg i} \end{bmatrix} \hat{\pi}_i, \tag{7.3}$$

where $\hat{D}_{\neg i} = \operatorname{diag}(\hat{A}_{\neg i}) \succ O$, and $\hat{A}_{\neg i}$ is the $(n - \hat{n}_i) \times (n - \hat{n}_i)$ complementary principal submatrix of $A$ as in (2.9). As in (2.13), we have here also the fundamental identity

$$\hat{E}_i \hat{M}_i^{-1} = \hat{R}_i^T \hat{A}_i^{-1} \hat{R}_i, \quad i = 1, \ldots, n.$$

We want to compare $\hat{M}_i$ with $M_i$, although $\hat{A}_i$ and $A_i$ are of different size. Without loss of generality, we can assume that the permutations $\pi_i$ and $\hat{\pi}_i$ coincide on the set $S_i$, and that the indexes in $S_i$ are the first $n_i$ elements in $\hat{S}_i$. In fact, we can assume that $\hat{\pi}_i = \pi_i$. Thus, $A_i$ is a principal submatrix of $\hat{A}_i$, and $\hat{M}_i$ has the same diagonal as $M_i$.

We will apply to these the following result for symmetric positive definite matrices which can be found, e.g., in [21].

LEMMA 7.1. [illegible] $A$ [illegible] $\tilde{M}_i$ [illegible] (6.4) [illegible] $A$ [illegible] $A_i = R_i A R_i^T$ $R_i$ [illegible] $A_i$ [illegible] $A$ [illegible] $R_i^T A_i^{-1} R_i \preceq A^{-1}$

We consider the case of damped additive Schwarz with iteration matrix (2.4), and the iteration matrix corresponding to the larger overlap is

$$\hat{T}_{AS,\theta} = I - \theta \sum_{i=1}^{p} \hat{R}_i^T \hat{A}_i^{-1} \hat{R}_i A. \tag{7.4}$$

THEOREM 7.2. [illegible] $A$ [illegible] $b \in \mathcal{R}(A)$ [illegible] $x_0 \notin \mathcal{N}(A)$ [illegible] $A$ [illegible] (7.1) [illegible] (2.4) [illegible] (7.4) [illegible] $\theta \leq 1/p$ [illegible] $\gamma(\hat{T}_\theta) \leq \gamma(T_\theta)$

[illegible] As mentioned above assume that all the principal submatrices of $A$ of order less than $n$ are nonsingular. Let $Q_i = E_i M_i^{-1} = R_i^T A_i^{-1} R_i$ and $\hat{Q}_i = \hat{E}_i \hat{M}_i^{-1} = \hat{R}_i^T \hat{A}_i^{-1} \hat{R}_i$. Since $A_i$ is a principal submatrix of $\hat{A}_i$, by Lemma 7.1 we have that $\hat{Q}_i \succeq Q_i$. Therefore,

$$\hat{M}_\theta^{-1} = \theta \sum_{i=1}^{p} \hat{Q}_i \succeq \theta \sum_{i=1}^{p} Q_i = M_\theta^{-1} \succ O.$$

As shown in Remark 4.3, these splittings are strong $P$-regular, and the theorem follows from Theorem 3.6. □

We note that an alternative proof similar to that in Remark 6.2 can be applied here, using the relation $\hat{R}_i^T \hat{A}_i^{-1} \hat{R}_i = \hat{Q}_i \succeq Q_i = R_i^T A_i^{-1} R_i$ just proved.

Theorem 7.2 indicates that the more overlap there is, the faster the convergence of the algebraic additive Schwarz method. As a special case, we have that overlap is better than no overlap. This is consistent with the analysis for grid-based methods; see, e.g., [4], [41]. Of course, the faster convergence rate brings an associated increased cost of the local solvers, since now they have matrices of larger dimension and more nonzeros. In the cited references a small amount of overlap is recommended, and the increase in cost is usually offset by faster convergence.

We should mention that with an increase of overlap, the number of colors of the graph may decrease, so that the damping factor may need to be revised. In all cases, the maximum restriction is $\theta < 1/p$.

A comparison analogous to that of Theorem 7.2 is not valid for multiplicative Schwarz, not even in the definite case. A counterexample can be found in [40].

## 8. Varying the number of blocks.

We address here the following question: If we partition a block into smaller blocks, how is the convergence of the Schwarz method affected? We show that for the additive Schwarz method the more subblocks (subdomains), the slower the convergence. In a limiting case, if we have a single variable in each block and there is no overlap, this is the classic Jacobi method, and our results indicate that this has asymptotically slower convergence than any sets of blocks for additive Schwarz.

As in the situations described in sections 6 and 7, the slower convergence may be partially compensated by less expensive local solvers, since they are of smaller dimension.

Formally, consider each block of variables $S_i$ partitioned into $k_i$ subblocks; i.e., we have

$$S_{i_j} \subset S_i, \quad j = 1, \dots, k_i, \tag{8.1}$$

$\bigcup_{j=1}^{k_i} S_{i_j} = S_i$, and $S_{i_j} \cap S_{i_k} = \emptyset$ if $j \neq k$. Each set $S_{i_j}$ has associated matrices $R_{i_j}$ and $E_{i_j} = R_{i_j}^T R_{i_j}$. Since we have a partition,

$$E_{i_j} \preceq E_i, \quad j = 1, \dots, k_i, \quad \text{and} \quad \sum_{j=1}^{k_i} E_{i_j} = E_i, \ i = 1, \dots, p. \tag{8.2}$$

We define the matrices $A_{i_j} = R_{i_j} A R_{i_j}^T$, and $M_{i_j}$ corresponding to the set $S_{i_j}$ in the manner already familiar to the reader (see, e.g., (7.3)), so that

$$E_{i_j} M_{i_j}^{-1} = R_{i_j}^T A_{i_j}^{-1} R_{i_j}, \ j = 1, \dots, k_i, \ i = 1, \dots, p.$$

Given a fixed damping parameter $\theta$, the iteration matrix of the refined partition is then

$$\bar{T}_\theta = I - \theta \sum_{i=1}^{p} \sum_{j=1}^{k_i} E_{i_j} M_{i_j}^{-1} A \tag{8.3}$$

(cf. (2.4)), and an induced strong $P$-splitting (assuming the proper restriction on $\theta$) $A = \bar{M}_\theta - \bar{N}_\theta$ is given by

$$\bar{M}_\theta^{-1} = \theta \sum_{i=1}^{p} \sum_{j=1}^{k_i} E_{i_j} M_{i_j}^{-1}.$$

THEOREM 8.1. [illegible] $A$ [illegible] $b \in \mathcal{R}(A)$ [illegible] $x_0 \notin \mathcal{N}(A)$ [illegible] $A$ [illegible] (2.11) [illegible] (8.1) [illegible] (2.4) [illegible] (8.3) [illegible] $k = \max_i k_i$ [illegible] $\theta \leq 1/p$ [illegible] $\bar{\theta} = \theta/k \leq 1/(kp)$ [illegible] $\gamma(T_\theta) \leq \gamma(\bar{T}_{\bar{\theta}})$

[illegible]. As in the proof of Theorem 7.2 we have, using Lemma 7.1, that

$$Q_{i_j} = E_{i_j} M_{i_j}^{-1} \preceq Q_i = E_i M_i^{-1}.$$

Therefore, $\sum_{j=1}^{k_i} Q_{i_j} \preceq k_i Q_i$, and

$$\bar{M}_\theta^{-1} = \theta \sum_{i=1}^{p} \sum_{j=1}^{k_i} Q_{i_j} \preceq k\theta \sum_{i=1}^{p} Q_i = k M_\theta^{-1},$$

which is equivalent to $\bar{M}_{\bar{\theta}}^{-1} = (1/k)\bar{M}_\theta^{-1} \preceq M_\theta^{-1}$. The theorem now follows using Theorem 3.6 and the fact that these are strong $P$-regular splittings, as shown in Remark 4.3. ☐

As in the previous sections a comparison analogous to that of Theorem 8.1 is not valid for multiplicative Schwarz, not even in the definite case. Again, a counterexample can be found in [40].

**9. Two-level schemes.** We consider now two-level schemes, i.e., those in which an additional step is taken, corresponding to a coarse grid correction. In the nonsingular case, this additional step makes Schwarz methods optimal in the sense that the condition number of the preconditioned matrix $M^{-1}A$ is independent of the mesh size; see, e.g., [38], [41], [44]. In our setting, for the coarse grid correction consider an additional subspace $V_0$ of $V$, and the corresponding projection $P_0 = R_0^T A_0^{-1} R_0 A = R_0^T (R_0 A R_0^T)^{-1} R_0 A$. There are several cases we consider here: additive Schwarz with coarse grid correction, with iteration matrix given by

$$T_{ASc,\theta} = T_{AS,\theta} - \theta R_0^T A_0^{-1} R_0 A = I - \theta \sum_{i=0}^{p} R_i^T A_i^{-1} R_i A = I - \theta \sum_{i=0}^{p} P_i; \tag{9.1}$$

multiplicative Schwarz with coarse grid correction, with iteration matrix given by

$$T_{MSc} = T_{MS}(I - P_0) = \prod_{i=p}^{0} (I - P_i),$$

or in the symmetrized case by $T_{SMSc} = (I - P_0) T_{SMS} (I - P_0)$; multiplicative Schwarz additively corrected, known as the two-level hybrid I Schwarz method, with iteration matrix given by

$$H_{I,\theta} = I - \theta P_0 - \theta(I - T_{MS}) = I - \theta(G_0 + M_{MS}^{-1})A,$$

where $G_0 = R_0^T A_0^{-1} R_0$; and the two-level hybrid II Schwarz method, which is additive Schwarz multiplicatively corrected, with iteration matrix given by

$$H_{II,\theta} = T_{AS,\theta}(I - P_0).$$

We begin our analysis with the additive Schwarz iteration with coarse grid correction. By comparing the iteration matrices in (9.1) and (2.4), one can see that

Theorem 4.2 is valid in this case as well, with the exception that the damping factor $\theta$ needs to be less than $2/(p+1)$. Therefore we have that the matrix $T_{ASc,\theta}$ is a convergent matrix, and that the induced splitting defined by $M_{ASc,\theta}^{-1} = \theta \sum_{i=0}^{p} R_i^T A_i^{-1} R_i$ is $P$-regular. We can also show that coarse grid correction does not increase (and may decrease) the convergence factor of the iterations.

THEOREM 9.1. [illegible] $A$ [illegible] $\gamma(T_{ASc,\theta}) \leq \gamma(T_{AS,\theta})$

[illegible]. We use the fact that $G_0 = R_0^T A_0^{-1} R_0 \succeq 0$ to conclude that

$$M_{ASc,\theta}^{-1} = \theta(M_{AS}^{-1} + G_0) \succeq \theta M_{AS}^{-1} \; .$$

The theorem now follows by the application of Theorem 3.6. ☐

A characterization similar to (4.4) applies to this two-level method, with one more term in the sum. Thus, an alternative proof of this theorem using this characterization can be done in a manner similar to that in Remark 6.2.

Next, we consider the multiplicative Schwarz iterations with coarse grid correction. It is not hard to see that Theorem 5.2 applies to this case as well, so that $T_{MSc}$ and $T_{SMSc}$ are convergent.

We conclude by mentioning that the coarse grid corrections can be applied to the methods with inexact solvers described in section 6 as well, and since the analysis is very similar, we do not repeat it.

**Acknowledgments.** We thank Michele Benzi and the referees for their comments on an earlier version of this paper, which helped improve our presentation.

## REFERENCES

[1] M. Benzi, A. Frommer, R. Nabben, and D. B. Szyld, *Algebraic theory of multiplicative Schwarz methods*, Numer. Math., 89 (2001), pp. 605–639.

[2] M. Benzi and D. B. Szyld, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–321.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[4] P. E. Bjørstad and O. B. Widlund, *To overlap or not to overlap: A note on a domain decomposition method for elliptic problems*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 1053–1061.

[5] P. Bochev and R. B. Lehoucq, *On the finite element solution of the pure Neumann problem*, SIAM Rev., 47 (2005), pp. 50–66.

[6] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of non-overlapping domain decomposition algorithms with inexact solves*, Math. Comp., 67 (1998), pp. 1–19.

[7] R. Bru, F. Pedroche, and D. B. Szyld, *Additive Schwarz iterations for Markov chains*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 445–458.

[8] T. F. Chan and T. P. Mathew, *Domain decomposition algorithms*, in Acta Numerica 1994, Acta Numer., Cambridge University Press, Cambridge, UK, 1994, pp. 61–143.

[9] J.-J. Climent and C. Perea, *Some comparison theorems for weak nonnegative splittings of bounded operators*, Linear Algebra Appl., 275–276 (1998), pp. 77–106.

[10] G. Csordas and R. S. Varga, *Comparisons of regular splittings of matrices*, Numer. Math., 44 (1984), pp. 23–35.

[11] M. Dryja, *An additive Schwarz algorithm for two- and three-dimensional finite element problems*, in Proceedings of the Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, G. A. Meurant, J. Pèriaux, and O. B. Widlund, eds., SIAM, Philadelphia, 1989, pp. 168–172.

[12] M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Technical Report 339, Ultracomputer Note 131, Department of Computer Science, Courant Institute, New York University, New York, NY, 1987.

[13] M. Dryja and O. B. Widlund, *Some domain decomposition algorithms for elliptic problems*, in Iterative Methods for Large Linear Systems, Academic Press, San Diego, 1989, pp. 273–291.

[14] M. Dryja and O. B. Widlund, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia, 1990, pp. 3–21.

[15] M. Dryja and O. B. Widlund, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math., 48 (1995), pp. 121–155.

[16] L. Elsner, *Comparisons of weak regular splittings and multisplitting methods*, Numer. Math., 56 (1989), pp. 283–289.

[17] R. W. Freund and M. Hochbruck, *On the use of two QMR algorithms for solving singular systems and applications in Markov Chain modelling*, Numer. Linear Algebra Appl., 1 (1994), pp. 403–420.

[18] A. Frommer and D. B. Szyld, *Weighted max norms, splittings, and overlapping additive Schwarz iterations*, Numer. Math., 83 (1999), pp. 259–278.

[19] A. Frommer and D. B. Szyld, *An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms*, SIAM J. Numer. Anal., 39 (2001), pp. 463–479.

[20] M. Griebel and P. Oswald, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numer. Math., 70 (1995), pp. 163–180.

[21] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Springer-Verlag, New York, 1994.

[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[23] W. J. Kammerer and M. Z. Nashed, *On the convergence of the conjugate gradient method for singular linear operator equations*, SIAM J. Numer. Anal., 9 (1972), pp. 165–181.

[24] H. B. Keller, *On the solution of singular and semidefinite linear systems by iteration*, SIAM J. Numer. Anal., 2 (1965), pp. 281–290.

[25] A. Klawonn and O. B. Widlund, *A domain decomposition method with Lagrange multipliers and inexact solvers for linear elasticity*, SIAM J. Sci. Comput., 22 (2000), pp. 1199–1219.

[26] Y.-J. Lee, J. Wu, J. Xu, and L. Zikatanov, *A Sharp Convergence Estimate of the Method of Subspace Corrections for Singular Systems*, Technical Report AM259, Department of Mathematics, Pennsylvania State University, State College, PA, 2002.

[27] Y.-J. Lee, J. Wu, J. Xu, and L. Zikatanov, *On the convergence of iterative methods for semidefinite linear systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 634–641.

[28] Y.-J. Lee, J. Xu, and L. Zikatanov, *Successive subspace correction method for singular system of equations*, in Proceedings of the Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., UNAM Press, Mexico City, Mexico, 2003, pp. 315–321. Also available online at http://www.ddm.org.

[29] I. Marek and D. B. Szyld, *Comparison theorems for weak splittings of bounded operators*, Numer. Math., 58 (1990), pp. 387–397.

[30] I. Marek and D. B. Szyld, *Comparison theorems for the convergence factor of iterative methods for singular matrices*, Linear Algebra Appl., 316 (2000), pp. 67–87.

[31] I. Marek and D. B. Szyld, *Comparison of convergence of general stationary iterative methods for singular matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 68–77.

[32] I. Marek and D. B. Szyld, *Algebraic Schwarz methods for the numerical solution of Markov chains*, Linear Algebra Appl., 386 (2004), pp. 67–81.

[33] R. Nabben, *A note on comparison theorems of splittings and multisplittings of Hermitian positive definite matrices*, Linear Algebra Appl., 233 (1996), pp. 67–80.

[34] R. Nabben, *Comparisons between additive and multiplicative Schwarz iterations in domain decomposition methods*, Numer. Math., 95 (2003), pp. 145–162.

[35] R. Nabben and D. B. Szyld, *Convergence theory of restricted multiplicative Schwarz methods*, SIAM J. Numer. Anal., 40 (2003), pp. 2318–2336.

[36] J. M. Ortega, *Numerical Analysis: A Second Course*, Classics Appl. Math. 3, SIAM, Philadelphia, 1990.

[37] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics Appl. Math. 30, SIAM, Philadelphia, 2000.

[38] A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Clarendon Press, Oxford, UK, 1999.

[39] L. Reichel and Q. Ye, *Breakdown-free GMRES for singular systems*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1001–1021.

[40] M. Schnitker, *Eine algebraische Konvergenztheorie der Schwarz-Verfahen für symmetrisch positiv definite Matrizen*, Examensarbeit, Universität Bielefeld, Bielefeld, Germany, 2002.

[41] B. F. Smith, P. E. Bjørstad, and W. D. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.

[42] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

[43] D. B. Szyld, *Equivalence of convergence conditions for iterative methods for singular equations*, Numer. Linear Algebra Appl., 1 (1994), pp. 151–154.

[44] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Series in Computational Mathematics 34, Springer-Verlag, Berlin, 2005.

[45] R. S. Varga, *Matrix Iterative Analysis. Second Revised and Expanded Edition*, Springer-Verlag, Berlin, 2000.

[46] Z. I. Woźnicki, *Nonnegative splitting theory*, Japan J. Indust. Appl. Math., 11 (1994), pp. 289–342.

[47] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 117–142

# MATRIX MEASURES AND RANDOM WALKS WITH A BLOCK TRIDIAGONAL TRANSITION MATRIX[*]

HOLGER DETTE[†], BETTINA REUTHER[†], W. J. STUDDEN[‡], AND M. ZYGMUNT[§]

**Abstract.** In this paper we study the connection between matrix measures and random walks with a block tridiagonal transition matrix. We derive sufficient conditions such that the blocks of the $n$-step block tridiagonal transition matrix of the Markov chain can be represented as integrals with respect to a matrix valued spectral measure. Several stochastic properties of the processes are characterized by means of this matrix measure. In many cases this measure is supported in the interval $[-1, 1]$. The results are illustrated by several examples including random walks on a grid and the embedded chain of a queuing system.

**Key words.** Markov chain, block tridiagonal transition matrix, spectral measure, matrix measure, quasi-birth-and-death process, canonical moments, Chebyshev matrix polynomials

**AMS subject classifications.** 60J10, 42C05

**DOI.** 10.1137/050638230

## 1. Introduction.

Consider a homogeneous Markov chain with state space

$$\mathcal{C}_d = \{(i, j) \in \mathbb{N}_0 \times \mathbb{N} \mid 1 \leq j \leq d\} \tag{1.1}$$

and block tridiagonal transition matrix

$$P = \begin{pmatrix} B_0 & A_0 & & & 0 \\ C_1^T & B_1 & A_1 & & \\ & C_2^T & B_2 & A_2 & \\ 0 & & \ddots & \ddots & \ddots \end{pmatrix}, \tag{1.2}$$

where $d \in \mathbb{N}$ is finite, and $A_0, A_1, \ldots, B_0, B_1, \ldots, C_1, C_2, \ldots$ are $d \times d$ matrices containing the probabilities of one-step transitions (here and throughout this paper $C^T$ denotes the transpose of the matrix $C$). If the one-step block tridiagonal transition matrix is represented by

$$P = (P_{ii'})_{i,i'=0,1,\ldots} \tag{1.3}$$

with $d \times d$ block matrices $P_{ii'}$, the probability of going in one step from state $(i, j)$ to $(i', j')$ is given by the element in the position $(j, j')$ of the matrix $P_{ii'}$. In the state $(i, j)$, $i$ is usually referred to as the level of the state and $j$ is referred to as the phase

---

[*] Received by the editors August 16, 2005; accepted for publication (in revised form) by M. Benzi June 22, 2006; published electronically December 21, 2006.
http://www.siam.org/journals/simax/29-1/63823.html

[†] Ruhr-Universität Bochum, Fakultät für Mathematik, 44780 Bochum, Germany (holger.dette@rub.de, bettina.reuther@rub.de). The work of the first author was supported by the Deutsche Forschungsgemeinschaft (De 502/22-1, SFB 475, Komplexitätsreduktion in multivariaten Datenstrukturen).

[‡] Department of Statistics, Purdue University, West Lafayette, IN 47907-1399 (studden@stat.purdue.edu).

[§] Department of Applied Mathematics, AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow, Poland (marcin.zygmunt@vp.pl).

of the state. Some illustrative examples will be given below. Block tridiagonal transition matrices of the form (1.2) naturally appear in the analysis of the embedded Markov chains of continuous-time Markov processes with state space (1.1) and block tridiagonal infinitesimal generator (see, e.g., the monographs of Neuts (1981) and Neuts (1989) or the recent work of Marek (2003) and Dayar and Quessette (2002) among many others) and these models have significant applications in the performance evalutation of communication systems (see, e.g., Ost (2001)). Markov chains with transition matrix (1.2) are known in the literature as level dependent quasi-birth-and-death processes and several authors have contributed to the analysis of such processes (see Hajek (1982), Gaver, Jacobs, and Latouche (1984), Ramaswami and Taylor (1996), Bright and Taylor (1995), Latouche, Pearce, and Taylor (1998), Bean, Pollett, and Taylor (2000), and Li and Cao (2004) among many others). Bright and Taylor (1995) considered the problem of calculating the equilibrium distribution of a quasi-birth-and-death process for finite dimensional block matrices, while Ramaswami and Taylor (1996) investigated level dependent processes with infinite dimensional blocks. Quasistationary distributions of these processes were considered by Bean, Pollett, and Taylor (2000). Latouche, Pearce, and Taylor (1998) discussed the existence and the form of invariant measures for quasi-birth-and-death processes. In the present paper we propose an alternative methodology for analyzing some level dependent quasi-birth-and-death processes which is based on some spectral analysis of the transition matrix.

For this we note that matrices of the form (1.2) are also closely related to a sequence of matrix polynomials recursively defined by

$$xQ_n(x) = A_nQ_{n+1}(x) + B_nQ_n(x) + C_n^TQ_{n-1}(x),\ n \in \mathbb{N}_0, \tag{1.4}$$

where $Q_{-1}(x) = 0$ and $Q_0(x) = I_d$ denotes the $d \times d$ identity matrix. If $A_n = C_{n+1}$ and $B_n$ is symmetric it follows that there exists a matrix measure $\Sigma = \{\sigma_{ij}\}_{i,j=1,\dots,d}$ on the real line (here $\sigma_{ij}$ are signed measures such that for any Borel set $A \subset \mathbb{R}$ the matrix $\Sigma(A)$ is nonnegative definite) such that the polynomials $Q_j(x)$ are orthonormal with respect to a left inner product, i.e.,

$$\langle Q_i, Q_j\rangle = \int_{\mathbb{R}} Q_i(x)d\Sigma(x)Q_j^T(x) = \delta_{ij}I_d \tag{1.5}$$

(see, e.g., Sinap and Van Assche (1996), or Duran (1995)). In recent years several authors have studied properties of matrix orthonormal polynomials (see, e.g., Rodman (1990), Duran and Van Assche (1995), Duran (1996, 1999), and Dette and Studden (2001) among many others).

In the present paper we are interested in the relation between Markov chains with state space $\mathcal{C}_d$ defined in (1.1) and block tridiagonal transition matrix (1.2) and the polynomials $Q_j(x)$ defined by the recursive relation (1.4). In the case $d = 1$ this problem has been studied extensively in the literature (see Karlin and McGregor (1959), Whitehurst (1982), Woess (1985), Van Doorn and Schrijner (1993, 1995), and Dette (1996) among many others), but the case $d > 1$ is more difficult, because in this case a system of matrix polynomials $\{Q_j(x)\}_{j\geq 0}$ satisfying a recurrence relation of the form (1.4) is not necessarily orthogonal with respect to an inner product induced by a matrix measure. In section 2 we characterize the transition matrices of the form (1.2) such that there exists an integral representation for the corresponding $n$-step transition probabilities in terms of the matrix measure and corresponding orthogonal matrix polynomials, i.e.,

$$P_{ij}^n\left(\int Q_j(x)d\Sigma(x)Q_j^T(x)\right)=\left(\int x^nQ_i(x)d\Sigma(x)Q_j^T(x)\right),$$

where $P_{ij}^n$ denotes the $d\times d$ block of the $n$-step block tridiagonal transition matrix $P^n$ in the position $(i,j)$. In other words, the element in the position $(k,l)$ of $P_{ij}^n$ is the probability of going in $n$ steps from state $(i,k)$ to $(j,l)$ and admitting an integral representation. We also derive a sufficient condition such that the spectral (matrix) measure $\Sigma$ (if it exists) is supported on the interval $[-1,1]$. In section 3 we discuss several illustrative examples where this condition is satisfied including some examples from queuing theory. Section 4 continues our more theoretical discussion and some consequences of the integral representation are derived. We present a characterization of recurrence by properties of the blocks of the transition matrix, which generalizes the classical characterization of recurrence of a birth-and-death chain (see Karlin and Taylor (1975)). Finally, in section 5 we present some applications of our results, which demonstrate the potential of our approach. In particular we derive a very simple necessary condition for positive recurrence of a quasi-birth-and-death process and a new representation of the equilibrium distribution in terms of the random walk measure $\Sigma$ and the orthogonal polynomials $Q_j(x)$.

## 2. Random walk matrix polynomials.

A matrix measure $\Sigma$ is a $d\times d$ matrix $\Sigma=\{\sigma_{ij}\}_{i,j=1,\dots,d}$ of finite signed measures $\sigma_{ij}$ on the Borel field of the real line $\mathbb{R}$ or of an appropriate subset. It will be assumed here that for each Borel set $A\subset\mathbb{R}$ the matrix $\Sigma(A)=\{\sigma_{ij}(A)\}_{i,j=1,\dots,d}$ is symmetric and nonnegative definite, i.e., $\Sigma(A)\geq 0$. The moments of the matrix measure $\Sigma$ are given by the $d\times d$ matrices

$$S_k=\int t^k d\Sigma(t),\quad k=0,1,\dots, \tag{2.1}$$

and only measures for which all relevant moments exist will be considered throughout this paper. Let $G_i$ $(i=0,\dots,n)$ denote $d\times d$ matrices; then a matrix polynomial is defined by $P(t)=\sum_{i=0}^n G_it^i$. The inner product of two matrix polynomials, say, $P$ and $Q$, is defined by

$$\langle P,Q\rangle \;=\; \int P(t)\Sigma(dt)Q^T(t), \tag{2.2}$$

where $Q^T(t)$ denotes the transpose of the matrix $Q(t)$. Sinap and Van Assche (1996) call this the "left" inner product. Orthogonal polynomials are defined by orthogonalizing the sequence $I_p, tI_p, t^2I_p,\dots$ with respect to the above inner product. If $S_0,S_1,\dots$ is a given sequence of matrices such that the block Hankel matrices

$$\underline{H}_{2m}=\begin{pmatrix} S_0 & \cdots & S_m\\ \vdots & & \vdots\\ S_m & \dots & S_{2m}\end{pmatrix} \tag{2.3}$$

are positive definite, it is well known (see, e.g., Marcellán and Sansigre (1993)) that a matrix measure $\Sigma$ with moments $S_j$ $(j\in\mathbb{N}_0)$ and a corresponding infinite sequence of orthogonal matrix polynomials with respect to $d\Sigma(x)$ exist. Moreover, these matrix polynomials satisfy a three term recurrence relation.

Let $\{Q_j(x)\}_{j\geq 0}$ denote a sequence of matrix polynomials defined by the recurrence relationship (1.4), where the matrices $C_j$ $(j \in \mathbb{N})$ and $A_j$ $(j \in \mathbb{N}_0)$ in (1.2) are assumed to be nonsingular. The following results characterize the existence of a matrix measure $\Sigma$ such that the polynomials $Q_j(x)$ are orthogonal with respect to $d\Sigma(x)$ in the sense of (2.2).

THEOREM 2.1. [illegible] $A_n$ $(n \in \mathbb{N}_0)$[illegible] $C_n$ $(n \in \mathbb{N})$[illegible] (1.2)[illegible] $\Sigma$ [illegible] $\underline{H}_{2m}$ $(m \in \mathbb{N}_0)$ [illegible] $\{Q_n(x)\}_{n\in\mathbb{N}_0}$ [illegible] (1.4)[illegible] $d\Sigma(x)$[illegible] $\{R_n\}_{n\in\mathbb{N}_0}$ [illegible]

$$
\begin{aligned}
&R_nB_nR_n^{-1} \text{ [illegible] } \quad \forall\ n \in \mathbb{N}_0, \\
(2.4) \qquad & \\
&R_n^TR_n = C_n^{-1}\cdots C_1^{-1}(R_0^TR_0)A_0\cdots A_{n-1} \quad \forall\ n \in \mathbb{N}.
\end{aligned}
$$

[illegible] Assume that the polynomials $\{Q_n(x)\}_{n\in\mathbb{N}_0}$ are orthogonal with respect to the measure $d\Sigma(x)$, that is,

$$
\int_{\mathbb{R}} Q_i(x)d\Sigma(x)Q_j^T(x) = 0, \tag{2.5}
$$

whenever $i \neq j$ and

$$
\int_{\mathbb{R}} Q_i(x)d\Sigma(x)Q_i^T(x) = F_i > 0 \quad (i \in \mathbb{N}_0), \tag{2.6}
$$

where we use the notation $F_i > 0$ for a positive definite matrix $F_i \in \mathbb{R}^{d\times d}$ (the fact that the matrix $F_i$ is positive definite follows from a straightforward calculation using the assumption that $\underline{H}_{2m}$ is positive definite for all $m \in \mathbb{N}_0$). Define $R_n = F_n^{-1/2}$ and $\tilde{Q}_n(x) = R_nQ_n(x)$; then it is easy to see that the polynomials $\{\tilde{Q}_n(x)\}_{n\in\mathbb{N}_0}$ are orthonormal with respect to the measure $d\Sigma(x)$. Therefore it follows from Sinap and Van Assche (1996) that there exist $d\times d$ nonsingular matrices $\{D_n\}_{n\in\mathbb{N}}$ and symmetric matrices $\{E_n\}_{n\in\mathbb{N}_0}$ such that the recurrence relation

$$
x\tilde{Q}_n(x) = D_{n+1}\tilde{Q}_{n+1}(x) + E_n\tilde{Q}_n(x) + D_n^T\tilde{Q}_{n-1}(x) \tag{2.7}
$$

is satisfied for all $n \in \mathbb{N}_0$, $(\tilde{Q}_{-1}(x) = 0,\ \tilde{Q}_0(x) = R_0)$. On the other hand, we obtain from (1.4) and the representation $\tilde{Q}_n(x) = R_nQ_n(x)$ the recurrence relation

(2.8)
$$
x\tilde{Q}_n(x) = R_nA_nR_{n+1}^{-1}\tilde{Q}_{n+1}(x) + R_nB_nR_n^{-1}\tilde{Q}_n(x) + R_nC_n^TR_{n-1}^{-1}\tilde{Q}_{n-1}(x),
$$

and a comparison of (2.7) and (2.8) yields

$$
D_{n+1} = R_nA_nR_{n+1}^{-1}, \quad E_n = R_nB_nR_n^{-1}\ , \quad D_n^T = R_nC_n^TR_{n-1}^{-1}, \tag{2.9}
$$

where the matrix $E_n$ is symmetric. Now a straightforward calculation gives

$$R_n A_n R_{n+1}^{-1} = (R_{n+1} C_{n+1}^T R_n^{-1})^T = (R_n^T)^{-1} C_{n+1} R_{n+1}^T,$$

or equivalently

$$R_{n+1}^T R_{n+1} = C_{n+1}^{-1} (R_n^T R_n) A_n.$$

This yields by an induction argument

$$R_n^T R_n = C_n^{-1} \cdots C_1^{-1} R_0^T R_0 A_0 \cdots A_{n-1}, \;\; n \in \mathbb{N},$$

and proves the first part of Theorem 2.1.

For the converse assume that the relations in (2.4) are satisfied and consider the polynomials $\tilde{Q}_n(x) = R_n Q_n(x)$. These polynomials satisfy the recurrence relation (2.8) and from (2.4) it follows that the matrices

$$E_n = R_n B_n R_n^{-1}$$

are symmetric ($n \in \mathbb{N}_0$), while

$$D_{n+1} = R_n A_n R_{n+1}^{-1} = (R_{n+1} C_{n+1}^T R_n^{-1})^T$$

by the second assumption in (2.4). Therefore the recurrence relation for the polynomials $\tilde{Q}_n(x)$ is of the form (2.7) and by the discussion following Theorem 3.1 in Sinap and van Assche (1996) these polynomials are orthonormal with respect to a matrix measure $d\Sigma(x)$. This also implies the orthogonality of the polynomials $Q_n(x) = R_n^{-1}\tilde{Q}_n(x)$ with respect to the measure $d\Sigma(x)$.

Because the polynomials $\underline{Q}_n(t) = R_0^{-1} D_1 \cdots D_n \tilde{Q}_n(t)$ have leading coefficient $I_d$ we obtain that the matrix

$$\langle \underline{Q}_n, \underline{Q}_n \rangle = \int \underline{Q}_n(t) d\Sigma(t) \underline{Q}_n^T(t) = R_0^{-1} D_1 \cdots D_n D_n^T \cdots D_1^T (R_0^T)^{-1} \tag{2.10}$$

is nonsingular. On the other hand it follows from Dette and Studden (2001) that the left-hand side of (2.10) is equal to the Schur complement, say, $S_{2n} - S_{2n}^-$, of $S_{2n}$ in $\underline{H}_{2n}$. Because the matrix $\underline{H}_{2n}$ is positive definite if and only if $\underline{H}_{2n-2}$ and the Schur complement of $S_{2n}$ in $\underline{H}_{2n}$ are positive definite it follows by an induction argument that all Hankel matrices obtained from the moments of the matrix measure $\Sigma$ are positive definite. $\square$

2.2. Throughout this paper a matrix measure $\Sigma$ with corresponding orthogonal matrix polynomials $Q_i(x)$ is called a random walk matrix measure or spectral measure and the polynomials $Q_i(x)$ will be called random walk matrix polynomials if the assumptions of Theorem 2.1 are satisfied. Because the polynomials $\tilde{Q}_i(x) = R_i Q_i(x)$ defined in the proof of Theorem 2.1 are orthonormal with respect to the measure $d\Sigma(x)$ it follows that

$$I_d = \langle \tilde{Q}_0, \tilde{Q}_0 \rangle = \int \tilde{Q}_0(x) d\Sigma(x) \tilde{Q}_0^T = R_0 S_0 R_0^T, \tag{2.11}$$

or equivalently

$$R_0^{-1}((R_0^T)^{-1}) = (R_0^T R_0)^{-1} = S_0, \tag{2.12}$$

where $S_0$ is the 0th moment of the matrix measure $\Sigma$ (see (2.1)). We finally note that the matrices $R_n$ in Theorem 2.1 are not unique. If $\{R_n\}_{n\in\mathbb{N}_0}$ is a sequence of matrices satisfying (2.4), these relations are also fulfilled for the sequence $\{\tilde{R}_n\}_{n\in\mathbb{N}_0} = \{U_nR_n\}_{n\in\mathbb{N}_0}$, where $U_n$ $(n \in \mathbb{N}_0)$ are arbitrary orthogonal matrices.

Before we present some examples, where the conditions of Theorem 2.1 are satisfied we derive some consequences of the existence of a random walk measure. For this let $Q(x) = (Q_0^T(x), Q_1^T(x), \dots)^T$ denote the vector of matrix polynomials defined by the recursive relation (1.4); then it is easy to see that the recurrence relation (1.4) is equivalent to

$$xQ(x) = PQ(x), \tag{2.13}$$

which gives (by iteration)

$$x^nQ(x) = P^nQ(x). \tag{2.14}$$

Therefore

$$\int x^nQ(x)d\Sigma(x)Q_j^T(x) = P^n\int Q(x)d\Sigma(x)Q_j^T(x), \tag{2.15}$$

and from the orthogonality of the random walk polynomials we obtain the representation

$$P_{ij}^n = \left(\int x^nQ_i(x)d\Sigma(x)Q_j^T(x)\right)\left(\int Q_j(x)d\Sigma(x)Q_j^T(x)\right)^{-1} \tag{2.16}$$

for the block in the position $(i,j)$ of the $n$-step block tridiagonal transition matrix $P^n$.

THEOREM 2.3. [illegible] 2.1 [illegible] $P_{ij}^n$ [illegible] $(i,j)$ [illegible] $n$ [illegible] $P^n$ [illegible] (2.16) [illegible] $\Sigma$ [illegible] $P$ [illegible]

[illegible] 2.4. Note that the random walk measure is not necessarily uniquely determined by the random walk on the grid $\mathcal{C}_d$. However, using the case $i = j = 0$ in (2.16) it follows for the moments of the random walk measure

$$P_{00}^n = S_nS_0^{-1} \quad (n \in \mathbb{N}_0), \tag{2.17}$$

where $P_{00}^n$ is the first block in the $n$-step transition matrix of the random walk. Therefore the moments of a random walk measure are essentially uniquely determined. In the following we will derive a sufficient condition such that the random walk measure (if it exists) is supported on the interval $[-1, 1]$. In this case the measure is determined by its moments.

THEOREM 2.5. [illegible] 2.1 [illegible] $R = \operatorname{diag}(R_0, R_1, R_2, \dots)$. [illegible] $R$ [illegible]

$$\tilde{P} = R^TPR^{-1} \tag{2.18}$$

[illegible] $\Sigma = \{\sigma_{ij}\}_{i,j=1,\dots,d}$ [illegible] (1.4) [illegible] $[-1,1]$, [illegible]

$$\operatorname{supp}(\sigma_{ij}) \subset [-1,1] \quad \forall\, i,j = 1,\dots,d.$$

[illegible] Note that the matrix in (2.18) is symmetric (because the assumptions of Theorem 2.1 are satisfied) and that the entries of $\tilde{P}$ are nonnegative, by the assumptions of the theorem. According to Schur's test (see Halmos and Sunder (1978), Theorem 5.2) it follows that

$$\|\tilde{P}\|_2 \le 1 \tag{2.19}$$

if we can find two vectors, say, $v, w$, with positive components such that

$$\tilde{P}v \le w \quad \text{and} \quad \tilde{P}w \le v$$

(where the symbol $\le$ means here inequality in each component). If $v = w = R1$ (here 1 denotes the infinite dimensional vector with all elements equal to one), then the representation (2.18) implies that

$$\tilde{P}v = \tilde{P}R1 = R^T P1 \le R^T 1,$$

which shows that (2.19) is indeed satisfied. Now let

$$\Pi_j = C_j^{-1} \dots C_1^{-1} R_0^T R_0 A_0 \dots A_{j-1} = R_j^T R_j, \tag{2.20}$$

and consider the inner product

$$\langle x, y\rangle_\Pi = \sum_{j=0}^{\infty} x_j^T \Pi_j y_j \tag{2.21}$$

(with $x = (x_0^T, x_1^T, \dots); y = (y_0^T, y_1^T, \dots); x_j \in \mathbb{R}^d, y_j \in \mathbb{R}^d$) and its corresponding norm, say, $\|\cdot\|_\Pi$. Define

$$\ell^2(\mathbb{R}^d) = \{x = (x_0^T, x_1^T, \dots) \mid x_j \in \mathbb{R}^d\ (j \in \mathbb{N}_0); \|x\|_\Pi^2 < \infty\}. \tag{2.22}$$

From the definition of $P$ and $\Pi_j$ it is easy to see that $\Pi_i P_{ij} = P_{ji}^T \Pi_j$ (for all $i, j \in \mathbb{N}_0$), which implies that $P$ is a selfadjoint operator with respect to the inner product $\langle\cdot,\cdot\rangle_\Pi$. Moreover, we have for any $x$

$$\begin{aligned} \|Px\|_\Pi &= x^T P^T \Pi P x = x^T R^T \tilde{P}^T \tilde{P} R x = \|\tilde{P} R x\|_2 \\ &\le \|\tilde{P}\|_2 \|Rx\|_2 \le x^T R^T R x = x^T \Pi x = \|x\|_\Pi, \end{aligned}$$

where we used the representation $\Pi = R^T R$ and (2.19). Consequently, $\|P\|_\Pi \le 1$, which proves the theorem. □

We note that there are many examples where the assumptions of Theorem 2.5 are satisfied and we conjecture, in fact, that a random walk measure is always supported in the interval $[-1,1]$. In the case $d = 1$ this property holds because in this case the assumptions of Theorems 2.1 and 2.5 are obviously satisfied. This was shown before by Karlin and McGregor (1959), and an alternative proof can be found in Dette and Studden (1997), Chapter 8.

Our next result gives a relation between the Stieltjes transforms of two random walk measures, say, $\Sigma$ and $\tilde{\Sigma}$, where only the matrices $B_0$ and $\tilde{B}_0$ differ in the corresponding one-step block tridiagonal transition matrices $P$ and $\tilde{P}$.

THEOREM 2.6. [illegible] $P$ [illegible] (1.2) [illegible]

$$\tilde{P} = \begin{pmatrix} \tilde{B}_0 & A_0 & & & 0 \\ C_1^T & B_1 & A_1 & & \\ & C_2^T & B_2 & A_2 & \\ & & & & \\ 0 & & & & \end{pmatrix}, \tag{2.23}$$

[illegible] $\Sigma$ [illegible] $P$ [illegible] $R_0\tilde{B}_0R_0^{-1}$ [illegible] $R_0$ [illegible] (2.4) [illegible] $\tilde{\Sigma}$ [illegible] $\tilde{P}$. [illegible] $\Sigma$ [illegible] $\tilde{\Sigma}$ [illegible]

$$\int \frac{d\Sigma(t)}{z-t} = \left\{ \left( \int \frac{d\tilde{\Sigma}(t)}{z-t} \right)^{-1} - S_0^{-1}(B_0 - \tilde{B}_0) \right\}^{-1}. \tag{2.24}$$

[illegible]. Because the matrix $R_0\tilde{B}_0R_0^{-1}$ is symmetric and the matrices $P$ and $\tilde{P}$ differ only by the element in the first block, the sequence of matrices $R_0, R_1, \dots$ can be used to symmetrize the matrices $P$ and $\tilde{P}$ simultaneously (see the proof of Theorem 2.1). Consequently, there exists a random walk measure corresponding to the random walk with one-step block tridiagonal transition matrix $\tilde{P}$. Let $\{Q_n(x)\}_{n\in\mathbb{N}_0}$ denote the system of matrix orthogonal polynomials defined by the recursive relation (1.4) and define $\{\tilde{Q}_n(x)\}_{n\in\mathbb{N}_0}$ by the same recursion, where the matrix $B_0$ has been replaced by $\tilde{B}_0$. A straightforward calculation shows that the difference polynomials

$$R_j(x) = \tilde{Q}_j(x) - Q_j(x)$$

also satisfy the recursion (1.4) with initial conditions $R_0(x) = 0$, $R_1(x) = A_0^{-1}(B_0 - \tilde{B}_0)$. In particular, these polynomials are "proportional" to the first associated orthogonal matrix polynomials

$$Q_n^{(1)}(x) = \int \frac{Q_n(x) - Q_n(t)}{x-t} d\Sigma(t) \quad (n \in \mathbb{N}_0), \tag{2.25}$$

that is,

$$R_n(x) = Q_n^{(1)}(x) R_0^T R_0 (B_0 - \tilde{B}_0). \tag{2.26}$$

Recall from the proof of Theorem 2.1 that the systems $\{R_nQ_n(x)R_0^{-1}\}_{n\in\mathbb{N}_0}$ and $\{R_n\tilde{Q}_n(x)R_0^{-1}\}_{n\in\mathbb{N}_0}$ are orthonormal with respect to the random walk measures $d\mu(x) = R_0 d\Sigma(x) R_0^T$ and $d\tilde{\mu} = R_0 d\tilde{\Sigma}(x) R_0^T$, respectively, and that $\mu$ and $\tilde{\mu}$ are determinate. Therefore we obtain from Markov's theorem for matrix orthogonal polynomials (see Duran (1996)) that

$$
\begin{aligned}
(2.27) \qquad \int \frac{d\tilde{\Sigma}(t)}{z-t} &= R_0^{-1} \int \frac{d\tilde{\mu}(t)}{z-t} (R_0^T)^{-1} \\
&= \lim_{n\to\infty} R_0^{-1} (R_n \tilde{Q}_n(z) R_0^{-1})^{-1} (R_n \tilde{Q}_n^{(1)}(z) R_0^T)(R_0^T)^{-1} \\
&= \lim_{n\to\infty} (\tilde{Q}_n(z))^{-1} \tilde{Q}_n^{(1)}(z) \\
&= \lim_{n\to\infty} \{Q_n(z) + Q_n^{(1)}(z) R_0^T R_0 (B_0 - \tilde{B}_0)\}^{-1} Q_n^{(1)}(z) \\
&= \lim_{n\to\infty} \{\{(Q_n(z))^{-1} Q_n^{(1)}(z)\}^{-1} + R_0^T R_0 (B_0 - \tilde{B}_0)\}^{-1} \\
&= \lim_{n\to\infty} \{R_0^T \{(R_n Q_n(z) R_0^{-1})^{-1} R_n Q_n^{(1)}(z) R_0^T\}^{-1} R_0 \\
&\qquad\qquad + R_0^T R_0 (B - \tilde{B}_0)\}^{-1} \\
&= \left\{ R_0^T \left( \int \frac{d\mu(t)}{z-t} \right)^{-1} R_0 + R_0^T R_0 (B_0 - \tilde{B}_0) \right\}^{-1} \\
&= \left\{ \left( \int \frac{d\Sigma(t)}{z-t} \right)^{-1} + R_0^T R_0 (B_0 - \tilde{B}_0) \right\}^{-1} \\
&= \left\{ \left( \int \frac{d\Sigma(t)}{z-t} \right)^{-1} + S_0^{-1} (B_0 - \tilde{B}_0) \right\}^{-1},
\end{aligned}
$$

where $\tilde{Q}_n^{(1)}(x)$ denotes the first associated orthogonal matrix polynomial obtained by the analogue of (2.25) from $\tilde{Q}_n(x)$ and we have used the fact that $\tilde{Q}_n^{(1)}(x) = Q_n^{(1)}(x)$ for the third equality (note that this identity is obvious from the definition of $P$ and $\tilde{P}$ in (1.2) and (2.23), respectively). ☐

[illegible] 2.7. Note that Theorem 2.1 and some of its consequences are derived under the assumption of nonsingular matrices $A_n$ and $C_n$. As pointed out by a referee there are several applications in queuing theory where these matrices do not have full rank (see Latouche and Ramaswami (1999)). In this remark we indicate how the nonsingularity assumptions regarding the matrices $C_n$ can be relaxed (note that this covers most of the commonly used queuing models). For this purpose we rewrite the conditions in Theorem 2.1 as

$$
(2.28) \qquad C_{n+1} R_{n+1}^T R_{n+1} = R_n^T R_n A_n \quad \forall\, n \in \mathbb{N}_0
$$

and

$$
(2.29) \qquad R_n B_n = E_n R_n \quad \forall\, n \in \mathbb{N}_0,
$$

for some sequence of symmetric matrices $(E_n)_{n\in\mathbb{N}_0}$. Note that the conditions (2.28) and (2.29) were derived in the proof of Theorem 2.1 (under the assumptions of Theorem 2.1 they are in fact equivalent to (2.4)). We will now demonstrate that these conditions are in fact sufficient for the proof of the existence of a random walk measure using some spectral theory of selfadjoint operators (see, e.g., Berezanskii (1968)). In the following we indicate how such a measure can be derived; further details can be found in Berezanskii (1968), pages 501–607.

For this purpose define

$$\Pi_j = R_j^T R_j \quad (j \in \mathbb{N}_0)$$

and consider the space $L(\mathbb{R}^d, L(\mathbb{R}^d, \mathbb{R}^d))$ which can be identified with

$$\ell^2(\mathbb{R}^{d\times d}) := \left\{ X = \left(X_0^T, X_1^T, \dots\right)^T \mid X_j \in \mathbb{R}^{d\times d}, \langle\langle X, X\rangle\rangle < \infty \right\},$$

where the matrix valued pseudo inner product is defined by ($Y = \left(Y_0^T, Y_1^T, \dots\right)^T$)

$$\langle\langle X, Y\rangle\rangle := \sum_{j=1}^{\infty} X_j^T \Pi_j Y_j.$$

Note that the space $\ell^2(\mathbb{R}^{d\times d})$ equipped with the inner product

$$\langle X, Y\rangle = \frac{1}{d}\,\mathrm{trace}\langle\langle X, Y\rangle\rangle$$

is a Hilbert space and isometric isomorph to the space $\ell^2(\mathbb{R}^d)$ defined in (2.22). Moreover, the matrix $P$ in (1.2) defines an operator acting on $\ell^2(\mathbb{R}^{d\times d})$ and $\ell^2(\mathbb{R}^d)$, denoted by $P$ and $J$, respectively; that is,

$$\begin{aligned} (PX)_n &= A_n X_{n+1} + B_n X_n + C_n^T X_{n-1} \quad (n \in \mathbb{N}_0, X_{-1} = 0), \\ (Jx)_n &= A_n x_{n+1} + B_n x_n + C_n^T x_{n-1} \quad (n \in \mathbb{N}_0, x_{-1} = 0). \end{aligned}$$

Note that (2.28) and (2.29) imply the symmetry conditions

$$P_{ij}^T \Pi_i = \Pi_j P_{ji} \tag{2.30}$$

(this follows by an elementary calculation), and recalling the definition of the inner product $\langle\cdot,\cdot\rangle_\Pi$ in (2.21) we obtain

$$\langle Jx, y\rangle_\Pi = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} (P_{ij} x_j)^T \Pi_i y_i = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} x_j^T \Pi_j P_{ji} y_i = \langle x, Jy\rangle_\Pi.$$

In other words, $J$ is a selfadjoint operator acting on $\ell^2(\mathbb{R}^d)$. Let $(E_\lambda)_\lambda$ denote the corresponding resolution of the identity (i.e., $J = \int \lambda E_\lambda$); then $(E_\lambda)_\lambda$ induces a resolution of the identity, say, $(\mathcal{E}_\lambda)_\lambda$, corresponding to the operator $P$ on $\ell^2(\mathbb{R}^{d\times d})$ in the following way:

$$(\mathcal{E}_\lambda U)x := E_\lambda(Ux), \ U \in \ell^2(\mathbb{R}^{d\times d}), \ x \in \mathbb{R}^d.$$

Now define $E^{(j)} = (0_d, \dots, 0_d, I_d, 0_d, \dots)^T \in \ell^2(\mathbb{R}^{d\times d})$ as the $j$th "unit" vector (here $0_d$ is the $d \times d$ matrix with all entries equal to 0) and the spectral measure by

$$\Sigma(\lambda) = \langle\langle E^{(0)}, \mathcal{E}_\lambda E^{(0)}\rangle\rangle \ \in \mathbb{R}^{d\times d};$$

then it follows by similiar arguments as in Berezanskii (1968), pages 562–565, that

$$\int Q_i(x)d\Sigma(x)Q_j^T(x) = \langle\langle E^{(i)}, E^{(j)}\rangle\rangle = \delta_{ij}\Pi_j,$$

where $\delta_{ij}$ denotes Kronecker's symbol. The same arguments as used in the derivation of Theorem 2.3 now imply

$$P_{ij}^n\Pi_j = \int x^n Q_i(x)d\Sigma(x)Q_j^T(x),$$

which is the statement of Theorem 2.3. Other results of this paper can be generalized in a similiar way. For example, Theorem 2.5 remains valid if there exists a matrix $\tilde{P}$ with nonnegative entries such that $\tilde{P}R = R^TP$. The details are omitted for the sake of brevity.

**3. Examples.** In this section we present several examples where the conditions of Theorem 2.1 are satisfied.

**3.1. Random walks on the integers.** Consider the classical random walk on $\mathbb{Z}$ (see, e.g., Feller (1950)) with one-step up, down, and holding transition probabilities $p_i$, $q_i$, and $r_i$ (respectively), where $p_i + q_i + r_i \le 1$, $i \in \mathbb{Z}$, where the strict inequality $p_i + q_i + r_i < 1$ is interpreted as a permanent absorbing state $i^*$, which can be reached from the state $i$ with probability $1 - p_i - q_i - r_i$. By the one-to-one mapping

$$\psi : \begin{cases} \mathbb{Z} \to \mathcal{C}_2, \\ i \to \begin{cases} (i,1) & \text{if } i \in \mathbb{N}_0, \\ (-i-1,2) & \text{else}, \end{cases} \end{cases}$$

this process can be interpreted as a process on the grid $\mathcal{C}_2$, where transitions from the first to the second row are only possible if the process is in state $(0,1)$. The transition matrix of this process is given by (1.2) with $2 \times 2$ blocks

$$B_0 = \begin{pmatrix} r_0 & q_0 \\ p_{-1} & r_{-1} \end{pmatrix}; \quad B_n = \begin{pmatrix} r_n & 0 \\ 0 & r_{-n-1} \end{pmatrix}; \tag{3.1}$$

$$A_n = \begin{pmatrix} p_n & 0 \\ 0 & q_{-n-1} \end{pmatrix}; \quad C_n^T = \begin{pmatrix} q_n & 0 \\ 0 & p_{-n-1} \end{pmatrix}. \tag{3.2}$$

It is easy to see that the conditions of Theorem 2.1 are satisfied with the matrices

$$R_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{\frac{q_0}{p_{-1}}} \end{pmatrix}, \quad R_n = \begin{pmatrix} \sqrt{\frac{p_0 \cdots p_{n-1}}{q_1 \cdots q_n}} & 0 \\ 0 & \sqrt{\frac{q_0 q_{-1} \cdots q_{-n}}{p_{-1} p_{-2} \cdots p_{-n-1}}} \end{pmatrix}, \tag{3.3}$$

and consequently, there exists a random walk matrix measure corresponding to this process, say, $\Sigma$, which is supported in the interval $[-1,1]$ (see Theorem 2.5). For the calculation of the Stieltjes transform of this measure we use Theorem 2.6 and obtain

$$\Phi(z) = \int \frac{d\Sigma(t)}{z-t} = \left\{\tilde{\Phi}^{-1}(z) - R_0^T R_0(B_0 - \tilde{B}_0)\right\}^{-1}. \tag{3.4}$$

Here $\tilde{\Phi}$ is the Stieltjes transform of a random walk measure $\tilde{\Sigma}$ with transition matrix (1.2), where the matrix $B_0$ in (3.1) has been replaced by

$$\tilde{B}_0 = \begin{pmatrix} r_0 & 0 \\ 0 & r_{-1} \end{pmatrix},$$

and the matrix $B_0 - \tilde{B}_0$ is given by

$$B_0 - \tilde{B}_0 = \begin{pmatrix} 0 & q_0 \\ p_{-1} & 0 \end{pmatrix}.$$

Note that the matrix $\tilde{\Phi}$ is diagonal and if $\tilde{\Phi}^+$ and $\tilde{\Phi}^-$ denote the corresponding diagonal elements, we obtain from (3.4) the representation

$$\begin{aligned} \Phi(z) = \int \frac{d\Sigma(t)}{z-t} &= \begin{pmatrix} 1/\tilde{\Phi}^+(z) & -q_0 \\ -q_0 & 1/\tilde{\Phi}^-(z) \end{pmatrix}^{-1} \\ &= \frac{1}{1-q_0^2\tilde{\Phi}^+(z)\tilde{\Phi}^-(z)} \begin{pmatrix} \tilde{\Phi}^+(z) & q_0\tilde{\Phi}^-(z)\tilde{\Phi}^+(z) \\ q_0\Phi^-(z)\tilde{\Phi}^+(z) & \tilde{\Phi}^-(z) \end{pmatrix}. \end{aligned}$$

In particular, for the classical random walk ($p_i = p,\ q_i = q,\ r_i = 0$ for all $i \in \mathbb{Z}$) we have

$$\tilde{\Phi}^+(z) = -\frac{z-\sqrt{z^2-4pq}}{2pq}, \quad \tilde{\Phi}^-(z) = \frac{p}{q}\Phi^+(z),$$

and a straightforward calculation gives the result

$$\Phi(z) = \begin{pmatrix} \frac{-1}{\sqrt{z^2-4pq}} & \frac{1}{2q}\left(1-\frac{z}{\sqrt{z^2-4pq}}\right) \\ \frac{1}{2q}\left(1-\frac{z}{\sqrt{z^2-4pq}}\right) & \frac{p}{q}\frac{-1}{\sqrt{z^2-4pq}} \end{pmatrix},$$

which was also obtained by Karlin and McGregor (1959) by a probabilistic argument.

**3.2. An example from queuing theory.** In a recent paper Dayar and Quessette (2002) considered a system of two independent queues, where queue 1 is an $M/M/1$ and queue 2 is an $M/M/1/d-1$. Both queues have a Poisson arrival process with rate $\lambda_i$ $(i = 1, 2)$ and exponential service distributions with rates $\mu_i$ $(i = 1, 2)$. It is easy to see that the embedded random walk corresponding to the quasi-birth-and-death process representing the length of queue 1 (which is unbounded) and the length of queue 2 (which varies between $0, 1, \ldots, d-1$) has a one-step transition matrix of the form (1.2), where the blocks $B_i, A_i$, and $C_i$ are given by

$$B_0 = \begin{pmatrix} 0 & \frac{\lambda_2}{\lambda_1+\lambda_2} & & & \\ \frac{\mu_2}{\gamma-\mu_1} & 0 & \frac{\lambda_2}{\gamma-\mu_1} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\mu_2}{\gamma-\mu_1} & 0 & \frac{\lambda_2}{\gamma-\mu_1} \\ & & & \frac{\mu_2}{\lambda_1+\mu_2} & 0 \end{pmatrix}, \tag{3.5}$$

$$B_i = \begin{pmatrix} 0 & \frac{\lambda_2}{\gamma-\mu_2} & & & \\ \frac{\mu_2}{\gamma} & 0 & \frac{\lambda_2}{\gamma} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{\mu_2}{\gamma} & 0 & \frac{\lambda_2}{\gamma} \\ & & & \frac{\mu_2}{\gamma-\lambda_2} & 0 \end{pmatrix}, \ i \geq 1, \tag{3.6}$$

$$A_0 = \begin{pmatrix} \frac{\lambda_1}{\lambda_1+\lambda_2} & & & & \\ & \frac{\lambda_1}{\gamma-\mu_1} & & & \\ & & \ddots & & \\ & & & \frac{\lambda_1}{\gamma-\mu_1} & \\ & & & & \frac{\lambda_1}{\lambda_1+\mu_2} \end{pmatrix}, \tag{3.7}$$

$$A_i = \begin{pmatrix} \frac{\lambda_1}{\gamma-\mu_2} & & & & \\ & \frac{\lambda_1}{\gamma} & & & \\ & & \ddots & & \\ & & & \frac{\lambda_1}{\gamma} & \\ & & & & \frac{\lambda_1}{\gamma-\lambda_2} \end{pmatrix}, \; i \geq 1, \tag{3.8}$$

and

$$C_i = \begin{pmatrix} \frac{\mu_1}{\gamma-\mu_2} & & & & \\ & \frac{\mu_1}{\gamma} & & & \\ & & \ddots & & \\ & & & \frac{\mu_1}{\gamma} & \\ & & & & \frac{\mu_1}{\gamma-\lambda_2} \end{pmatrix}, \; i \geq 1, \tag{3.9}$$

respectively, and $\gamma = \lambda_1 + \lambda_2 + \mu_1 + \mu_2$, $\lambda_1 < \mu_1$. A straightforward calculation shows that the assumptions of Theorem 2.1 are satisfied, where the matrices $R_n$ are diagonal and given by

$$R_0 = \operatorname{diag}\left( \frac{\sqrt{(\lambda_1+\lambda_2)\mu_2}}{\sqrt{(\gamma-\mu_1)\lambda_2}}, 1, \frac{\sqrt{\lambda_2}}{\sqrt{\mu_2}}, \frac{\lambda_2}{\mu_2}, \dots, \left(\frac{\sqrt{\lambda_2}}{\sqrt{\mu_2}}\right)^{d-3}, \frac{\sqrt{(\lambda_1+\mu_2)\lambda_2^{d-2}}}{\sqrt{(\gamma-\mu_1)\mu_2^{d-2}}} \right),$$

$$R_1 = \operatorname{diag}\left( \frac{\sqrt{\lambda_1(\gamma-\mu_2)\mu_2}}{\sqrt{\lambda_2(\gamma-\mu_1)\mu_1}}, \frac{\sqrt{\gamma\lambda_1}}{\sqrt{(\gamma-\mu_1)\mu_1}}, \dots, \frac{\sqrt{\gamma\lambda_1\lambda_2^{d-3}}}{\sqrt{(\gamma-\mu_1)\mu_1\mu_2^{d-3}}}, \frac{\sqrt{\lambda_1(\gamma-\lambda_2)\lambda_2^{d-2}}}{\sqrt{(\gamma-\mu_1)\mu_1\mu_2^{d-2}}} \right),$$

$$R_i = \left(\sqrt{\frac{\lambda_1}{\mu_1}}\right)^{i-1} R_1, \; i \geq 2.$$

It also follows from Theorem 2.5 that the corresponding random walk matrix measure is supported in the interval $[-1, 1]$.

**3.3. The simple random walk on the grid.** Consider the random walk on the grid $\mathcal{C}_d$, where the probabilities of going from state $(i,j)$ to $(i,j+1), (i,j-1), (i-1,j), (i+1,j)$ are given by $u, v, \ell, r$, respectively, where $u+v+\ell+r = 1$. In this case it follows that $A_i = rI_d$ $(i \geq 0)$, $C_i = \ell I_d$ $(i \geq 1)$,

$$B_i = \begin{pmatrix} 0 & u & & & & \\ v & 0 & u & & & \\ & v & 0 & u & & \\ & & \ddots & \ddots & \ddots & \\ & & & v & 0 & u \\ & & & & v & 0 \end{pmatrix}, \; i \geq 0, \tag{3.10}$$

and it is easy to see that the conditions of Theorem 2.1 are satisfied with

$$R_0 = \operatorname{diag}\left(1, \sqrt{\frac{u}{v}}, \sqrt{\frac{u^2}{v^2}}, \dots, \sqrt{\frac{u^{d-1}}{v^{d-1}}}\right), \ R_i = \left(\sqrt{\frac{r}{\ell}}\right)^i R_0, \ i \geq 1.$$

It now follows from Theorem 2.5 that the corresponding random walk matrix measure is supported in the interval $[-1, 1]$. For the identification of the Stieltjes transform of the spectral measure we note that the orthonormal polynomials defined by (2.7) have constant coefficients given by $D = D_n = \sqrt{r\ell} I_d$,

$$E = E_n = \begin{pmatrix} 0 & \sqrt{vu} & & & & \\ \sqrt{vu} & 0 & \sqrt{vu} & & & \\ & \sqrt{vu} & 0 & \sqrt{vu} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \sqrt{vu} & 0 & \sqrt{vu} \\ & & & & \sqrt{vu} & 0 \end{pmatrix}. \tag{3.11}$$

Therefore it follows from the work of Duran (1999) that the Stieltjes transform of the random walk measure is given by

$$\int \frac{d\Sigma(t)}{z-t} = \frac{1}{2r\ell}\Big\{zI_d - E - \Big\{(zI_d - E)^2 - 4r\ell I_d\Big\}^{1/2}\Big\}.$$

From the same reference we obtain that the support of the random walk measure is given by the set

$$\operatorname{supp}(\Sigma) = \{x \in \mathbb{R} \mid xI_d - E \text{ has an eigenvalue in } [-2\sqrt{r\ell}, 2\sqrt{r\ell}]\}. \tag{3.12}$$

It is well known (see Basilevsky (1983)) that the eigenvalues of the matrix $E$ in (3.11) are given by

$$2\sqrt{uv}\cos\left(\frac{j\pi}{d+1}\right), \quad j = 1, \dots, d,$$

with corresponding normalized eigenvectors

$$x_j = \sqrt{\frac{2}{d+1}}\left(\sin\left(\ell\frac{\pi j}{d+1}\right)\right)_{\ell=1}^d.$$

Therefore it follows from (3.12) that

$$\operatorname{supp}(\Sigma) = \left[-2\sqrt{r\ell} + 2\sqrt{uv}\cos\left(\frac{\pi d}{d+1}\right), 2\sqrt{r\ell} + 2\sqrt{uv}\cos\left(\frac{\pi}{d+1}\right)\right]$$

(note that $\operatorname{supp}(\Sigma) \subset [-1, 1]$). For the calculation of the random walk measure we determine the spectral decomposition of the matrix

$$\begin{aligned} -H(x) &= 4I_d - D^{-1/2}(xI_d - E)D^{-1}(xI_d - E)D^{-1/2} \\ &= \frac{1}{r\ell}\Big\{4r\ell I_d - (xI_d - E)^2\Big\}. \end{aligned}$$

The eigenvalues of this matrix are given by

$$\lambda_j(x) = \frac{1}{r\ell}\left\{4r\ell - \left(x - 2\sqrt{vu}\cos\left(\frac{\pi j}{d+1}\right)\right)^2\right\},$$

and by the results in Duran (1999) the weight of the matrix measure is given by

$$d\Sigma(x) = \frac{1}{2\pi\sqrt{r\ell}} U\Lambda(x)U^T dx,$$

where the matrix $\Lambda(x)$ is defined by

$$\Lambda(x) = \Big\{ \mathrm{diag}(\max(\lambda_1(x), 0), \ldots, \max(\lambda_d(x), 0)) \Big\}^{1/2},$$

and the elements of the matrix $U = \{u_{j\ell}\}_{j,\ell=1,\ldots,d}$ are given by

$$u_{j\ell} = \sqrt{\frac{2}{d+1}} \sin\left(\ell \frac{j\pi}{d+1}\right).$$

**3.4. Finite state spaces.** The assertions of section 2 remain correct for random walks on a finite grid, where the corresponding random walk measure has a finite support. As an example consider a random walk on the finite grid

$$\mathcal{C} = \mathcal{C}_{d,N} = \{(i,j) \in \mathbb{N}_0 \times \mathbb{N} \,|\, 0 \le i \le N-1, 1 \le j \le d\},$$

where the probabilities of going from state $(i,j)$ to $(i,j+1), (i,j-1), (i-1,j), (i+1,j)$ are given by $u, v, \ell$, and $r$, respectively, where $u+v+\ell+r = 1$. Then the transition matrix $P$ is given by the finite dimensional block tridiagonal matrix

$$P = \begin{pmatrix} B_0 & A_0 & & & 0 \\ C_1^T & B_1 & A_1 & & \\ & \ddots & \ddots & \ddots & \\ & & C_{N-2}^T & B_{N-2} & A_{N-2} \\ 0 & & & C_{N-1}^T & B_{N-1} \end{pmatrix}$$

with $A_i = rI_d,\ 0 \le i \le N-2,\ C_i = \ell I_d,\ 1 \le i \le N-1$, and matrices $B_i$ defined by (3.10). A straightforward calculation shows that the corresponding random walk matrix polynomials are given by

$$Q_n(x) = \left(\sqrt{\frac{\ell}{r}}\right)^n U_n\left(\frac{1}{2}\sqrt{\frac{r}{\ell}}A\right),$$

$n = 0, \ldots, N-1$, where $U_n(z)$ denotes the Chebyshev polynomial of the second kind and the matrix $A$ is given by

$$A = \frac{1}{r} \begin{pmatrix} x & -u & & & & \\ -v & x & -u & & & \\ & -v & x & -u & & \\ & & \ddots & \ddots & \ddots & \\ & & & -v & x & -u \\ & & & & -v & x \end{pmatrix}.$$

Moreover, observing the representations

$$U_N\left(\frac{z}{2}\right) = \prod_{j=1}^{N} \left(z - 2\cos\left(\frac{j\pi}{N+1}\right)\right), \quad \det A = \left(\frac{\sqrt{uv}}{r}\right)^d U_d\left(\frac{1}{2\sqrt{uv}}x\right),$$

we obtain that the zeros of the polynomials $Q_N(x)$ are given by

$$\lambda_{ij} = 2\left(\sqrt{uv}\cos\left(\frac{i\pi}{d+1}\right) + \sqrt{\ell r}\cos\left(\frac{j\pi}{N+1}\right)\right), \quad i = 1,\dots,d; j = 1,\dots,N.$$

In particular, it follows for the rate of convergence of the probability of no absorption that

$$P(X_n \in \mathcal{C}_{d,N} \mid X_0 = x) = O\left(2^n\left(\sqrt{uv}\cos\left(\frac{\pi}{d+1}\right) + \sqrt{\ell r}\cos\left(\frac{\pi}{N+1}\right)\right)^n\right).$$

**3.5. A random walk on a tree.** Consider a graph with $d$ rays which are connected at one point, the origin. On each ray the probability of moving away from the origin is $p$ and moving in one step toward the origin is $q$, where $p + q = 1$. From the origin the probability of going to the $i$th ray is $d_i > 0\,(i = 1,\dots,d)$ (see Figure 1, where the case $d = 4$ is illustrated). It is easy to see that this process corresponds to a random walk on the grid $\mathcal{C}_d$ with block tridiagonal transition matrix $P$ in (1.2), where $B_i = 0$ if $i \geq 1$, $C_i = qI_d$ for all $i \geq 1$, $A_0 = \operatorname{diag}(d_1, p, \dots, p)$, $A_i = pI_d$ for all $i \geq 1$, and

$$B_0 = \begin{pmatrix} 0 & d_2 & \cdots & \cdots & d_d \\ q & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ q & 0 & \cdots & \cdots & 0 \end{pmatrix},$$

where $\sum_{i=1}^d d_i = 1$. Moreover, this matrix clearly satisfies the assumptions of Theorem 2.1 with

$$R_0 = \operatorname{diag}\left(1, \sqrt{\frac{d_2}{q}}, \dots, \sqrt{\frac{d_d}{q}}\right), \quad R_1 = \operatorname{diag}\left(\sqrt{\frac{d_1}{q}}, \sqrt{\frac{d_2 p}{q^2}}, \dots, \sqrt{\frac{d_d p}{q^2}}\right),$$

and

$$R_i = \left(\sqrt{\frac{p}{q}}\right)^{i-1} R_1, \; i \geq 2.$$

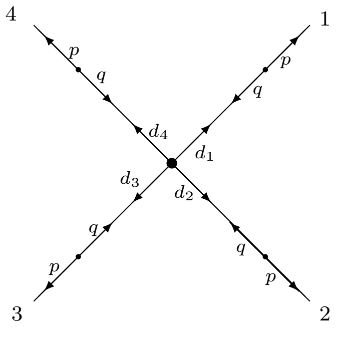


FIG. 1. *A random walk on a tree.*

By an application of Theorem 2.6 and the inversion formula for Stieltjes transforms we obtain for the corresponding random walk measure

$$d\Sigma(x) = \begin{bmatrix} a(x) & b_2(x) & b_3(x) & \dots & b_d(x) \\ b_2(x) & f_2(x) & e_{2,3}(x) & \dots & e_{2,d}(x) \\ b_3(x) & e_{2,3}(x) & f_3(x) & \dots & e_{3,d}(x) \\ \vdots & \vdots & \vdots & & \vdots \\ b_{d-1}(x) & e_{2,d-1}(x) & e_{3,d-1}(x) & \dots & e_{d-1,d}(x) \\ b_d(x) & e_{2,d}(x) & e_{3,d}(x) & \dots & f_d(x) \end{bmatrix} dx,$$

where the functions $a, b_i, e_{k,\ell}$, and $f_k$ are given by

$$a(x) = \frac{(\sum_{i=2}^d d_i^2 d_1 + d_1^2 q - (d_1 - p)x^2)\sqrt{4pq - x^2}}{2p\pi((\sum_{i=2}^d d_i^2 + d_1 q)^2 - (\sum_{i=2}^d d_i^2 + (d_1 - p)q)x^2)},$$

$$b_k(x) = -\frac{d_k x\sqrt{4pq - x^2}}{2\pi((\sum_{i=2}^d d_i^2 + d_1 q)^2 - (\sum_{i=2}^d d_i^2 + (d_1 - p)q)x^2)}, \quad k = 2, \dots, d,$$

$$e_{k,\ell}(x) = \frac{d_k d_\ell \sqrt{4pq - x^2}(px^2 - \sum_{i=2}^d d_j^2 d_1 - d_1^2 q)}{2\pi(d_1^2 q - (d_1 - p)x^2)((\sum_{i=2}^d d_i^2 + d_1 q)^2 - (\sum_{i=2}^d d_i^2 + (d_1 - p)q)x^2)},$$

$$k = 2, \dots, d-1,\ \ell = 3, \dots, d,$$

$$f_k(x) = \frac{d_1(\sum_{i=2}^d d_i^2 + d_1 q)(\sum_{i=2, i\neq k}^d d_i^2 + d_1 q)\sqrt{4pq - x^2}}{2\pi(d_1^2 q - (d_1 - p)x^2)((\sum_{i=2}^d d_i^2 + d_1 q)^2 - (\sum_{i=2}^d d_i^2 + (d_1 - p)q)x^2)}$$

$$+ \frac{-((\sum_{i=2}^d d_i^2 (d_1 - p) + \sum_{i=2, i\neq k}^d d_i^2 p + d_1(d_1 - p)q)x^2)\sqrt{4pq - x^2}}{2\pi(d_1^2 q - (d_1 - p)x^2)((\sum_{i=2}^d d_i^2 + d_1 q)^2 - (\sum_{i=2}^d d_i^2 + (d_1 - p)q)x^2)},$$

$$k = 2, \dots, d.$$

Note that the random walk measure is supported in the interval $[-2\sqrt{pq}, 2\sqrt{pq}]$.

## 4. Further discussion.

In the present section we derive further consequences of the existence of a random walk measure corresponding to the block tridiagonal transition matrix (1.2). Throughout this section we assume that the conditions of Theorem 2.1 are satisfied and that the corresponding random walk measure is supported in the interval $[-1, 1]$.

### 4.1. Recurrence.

We denote by

$$H_{ij}(z) = \sum_{n=0}^{\infty} (P_{ij}^n) z^n = \left( \int \frac{Q_i(x) d\Sigma(x) Q_j^T(x)}{1 - xz} \right) \left( \int Q_j(x) d\Sigma(x) Q_j^T(x) \right)^{-1} \tag{4.1}$$

the (matrix) generating function of the block $(i, j)$, where the last identity follows from Theorem 2.3 and Lebesgue's theorem. Therefore we obtain that a state $(i, \ell) \in \mathcal{C}_d$ is recurrent if and only if

$$\sum_{n=0}^{\infty} e_\ell^T P_{ii}^n e_\ell = \lim_{z\to 1} e_\ell^T H_{ii}(z) e_\ell = e_\ell^T \left(\int \frac{Q_i(x)d\Sigma(x)Q_i^T(x)}{1-x}\right)\left(\int Q_i(x)d\Sigma(x)Q_i^T(x)\right)^{-1} e_\ell = \infty, \tag{4.2}$$

where $e_\ell^T = (0,\dots,0,1,0,\dots,0)^T$ denotes the $\ell$th unit vector in $\mathbb{R}^d$. We summarize this observation in the following corollary.

Corollary 4.1. [illegible] 2.1 [illegible] $P$ [illegible] (1.2) [illegible] $\mathcal{C}_d$ [illegible] $[-1,1]$. [illegible] $(i,\ell)\in\mathcal{C}_d$ [illegible] (4.2) [illegible]

$$e_j^T \int_{-1}^{1} \frac{d\Sigma(x)}{1-x} S_0^{-1} e_j = \infty \tag{4.3}$$

[illegible] $j\in\{1,\dots,d\}$ [illegible] $j\in\{1,\dots,d\}$

Corollary 4.2. [illegible] 2.1 [illegible] $P$ [illegible] (1.2) [illegible] $\mathcal{C}_d$ [illegible] $[-1,1]$. [illegible] $d\tau_\ell(x) = e_\ell^T d\Sigma(x) S_0^{-1} e_\ell$ $(\ell = 1,\dots,d)$ [illegible] 1. [illegible] $d\tau_\ell(x)$ $(\ell = 1,\dots,d)$ [illegible] 1

[illegible]. Let $d\tau_\ell(x) = e_\ell^T d\Sigma(x) S_0^{-1} e_\ell$; then the probability of returning from state $(0,\ell)$ to $(0,\ell)$ in $k$ steps is given by

$$\alpha_k = e_\ell^T (P_{00}^k) e_\ell = e_\ell^T \int_{-1}^{1} x^k d\Sigma(x) S_0^{-1} e_\ell = \int_{-1}^{1} x^k d\tau_\ell(x).$$

The random walk is positive recurrent if and only if $\alpha = \lim_{k\to\infty}\alpha_k$ exists and is positive. Considering the sequence $\alpha_{2n}$ it follows by the dominated convergence theorem that this is the case if and only if $\tau_\ell$ has a jump at $x=-1$ or $x=1$. If $\tau_\ell$ has no jump at $x=1$ we obtain

$$\begin{aligned}\tau_\ell(-1) &= \lim_{n\to\infty}\left\{-\int_{-1}^{1} x^{2n+1} d\tau_\ell(x) + \int_{-1^-}^{1} x^{2n+1} d\tau_\ell(x)\right\}\\ &= -\lim_{n\to\infty} P_{00}^{2n+1} \le 0,\end{aligned}$$

and consequently $\tau_\ell$ has no jump at $x=-1$. Therefore the random walk is positive recurrent if and only if $\tau_\ell$ has a jump at $x=1$. □

[illegible] 4.3. For an irreducible random walk with a random walk measure $\Sigma$ satisfying $S_0 = I_d$ the properties of recurrence and positive recurrence are characterized by the diagonal elements of the corresponding random walk measure $\Sigma$.

**4.2. Canonical moments and random walk measures.** In this section we will represent the Stieltjes transform of a random walk matrix measure $\Sigma$ which is supported in the interval $[-1,1]$ in terms of its canonical moments, which were recently introduced by Dette and Studden (2001) in the context of matrix measures. We will use this representation to derive a characterization of recurrence of the process in terms of blocks of the matrix $P$.

THEOREM 4.4. [illegible] $\Sigma$, [illegible] $[-1,1]$ [illegible]

$$\int \frac{d\Sigma(x)}{z-x} = \lim_{n\to\infty} S_0^{1/2}\Big\{zI_d + I_d - 2\zeta_1^T - \Big\{zI_d + I_d - 2\zeta_2^T - 2\zeta_3^T - \Big\{zI_d + I_d - 2\zeta_4^T -2\zeta_5^T - \cdots - \Big\{zI_d + I_d - 2\zeta_{2n}^T - 2\zeta_{2n+1}^T\Big\}^{-1} 4\zeta_{2n}^T\zeta_{2n-1}^T\Big\}^{-1} \cdot \cdots 4\zeta_4^T\zeta_3^T\Big\}^{-1} 4\zeta_2^T\zeta_1^T\Big\}^{-1} S_0^{1/2}$$

$$= \lim_{n\to\infty} S_0^{1/2}\Big\{(z+1)I_d - \Big\{I_d - \Big\{(z+1)I_d - \cdots - \Big\{(z+1)I_d - 2\zeta_{2n+1}^T\Big\}^{-1} 2\zeta_{2n}^T\Big\}^{-1} \ldots 2\zeta_2^T\Big\}^{-1} 2\zeta_1^T\Big\}^{-1} S_0^{1/2},$$

[illegible] $\zeta_j \in \mathbb{R}^{d\times d}$ [illegible] $\zeta_0 = 0$ $\zeta_1 = U_1$, $\zeta_j = V_{j-1}U_j$ [illegible] $j \geq 2$ [illegible] $\{U_j\}$ [illegible] $\{V_j\}$ [illegible] $\Sigma$. [illegible] $\mathbb{C}$ [illegible] $[-1,1]$. [illegible]

$$\int \frac{d\Sigma(x)}{1-x} = \frac{1}{2} S_0^{1/2} \left[ I_d + \sum_{l=1}^{\infty} (V_1^T)^{-1} \ldots (V_l^T)^{-1} U_l^T \ldots U_1^T \right] S_0^{1/2}. \tag{4.4}$$

[illegible]. Let $\underline{P}_n(t)$ denote the $n$th monic orthogonal polynomial with respect to the matrix measure $d\Sigma(t)$; then it follows from Dette and Studden (2001) that $\underline{P}_n(t)$ can be calculated recursively as

$$\underline{P}_{n+1}(t) = \Big\{(t+1)I_d - 2\zeta_{2n+1}^T - 2\zeta_{2n}^T\Big\}\underline{P}_n(t) - 4\zeta_{2n}^T\zeta_{2n-1}^T\underline{P}_{n-1}(t), \tag{4.5}$$

where $\underline{P}_0(t) = I_d, \underline{P}_{-1}(t) = 0$, the quantities $\zeta_j \in \mathbb{R}^{d\times d}$ are defined by $\zeta_0 = 0$, $\zeta_1 = U_1$, $\zeta_j = V_{j-1}U_j$ if $j \geq 2$, and the sequences $\{U_j\}$ and $\{V_j\}$ are the canonical moments of the random walk measure $\Sigma$. Note that Dette and Studden (2001) define the canonical moments for matrix measures on the interval $[0,1]$, but the canonical moments are invariant with respect to transformations of the measure. More precisely, it can be shown that measures related by an affine transformation $t \to a + (b-a)t$ ($a, b \in \mathbb{R}$, $a < b$) have the same canonical moments. The results for the corresponding orthogonal polynomials can also easily be extended to matrix measures on the interval $[-1,1]$. The quantities

$$\Delta_{2n} := \langle \underline{P}_n, \underline{P}_n \rangle = 2^{2n}(S_0\zeta_1 \ldots \zeta_{2n})^T \tag{4.6}$$

are positive definite (see Dette and Studden (2001)) and consequently the polynomials

$$P_n(z) = \Delta_{2n}^{-1/2}\underline{P}_n(z)$$

are orthonormal with respect to the measure $d\Sigma(x)$. Now a straightforward calculation shows that these polynomials satisfy the recurrence relation

$$tP_k(t) = A_{k+1}P_{k+1}(t) + B_kP_k(t) + A_k^TP_{k-1}(t), \quad k = 0, 1, \ldots, \tag{4.7}$$

with initial conditions

$$P_{-1}(t) = 0,\ P_0(t) = S_0^{-1/2} \tag{4.8}$$

and coefficients

$$\begin{aligned} A_{n+1} &= \Delta_{2n}^{-1/2}\Delta_{2n+2}^{1/2}, && (4.9)\\ B_n &= -\Delta_{2n}^{-1/2}(I_d - 2\zeta_{2n}^T - 2\zeta_{2n+1}^T)\Delta_{2n}^{1/2}, && (4.10)\\ A_n^T &= 4\Delta_{2n}^{-1/2}\zeta_{2n}^T\zeta_{2n-1}^T\Delta_{2n-2}^{1/2} && (4.11) \end{aligned}$$

(note that the matrix $\Delta_{2n} = 4\Delta_{2n-2}\zeta_{2n-1}\zeta_{2n}$ is symmetric and therefore the two representations in (4.9) and (4.11) for the matrix $A_n$ are in fact identical). If $P_n^{(1)}(z)$ denotes the first associated orthogonal polynomial corresponding to $P_n(z)$ we obtain from Zygmunt (2002) the representation

$$\begin{aligned} F_n(z) &= (P_{n+1}(z))^{-1}P_{n+1}^{(1)}(z)\\ &= S_0\{zI_d - B_0 - A_1\{zI_d - B_1 - A_2\{zI_d - B_2 -\\ &\qquad \cdots - A_n\{zI_d - B_n\}^{-1}A_n^T\}^{-1}\ldots A_2^T\}^{-1}A_1^T\}^{-1}. \end{aligned} \tag{4.12}$$

Now a straightforward application of (4.9)–(4.11) yields

$$\begin{aligned} F_n(z) = S_0^{1/2}\{&zI_d + I_d - 2\zeta_1^T - \{zI_d + I_d - 2\zeta_2^T - 2\zeta_3^T - \{zI_d + I_d - 2\zeta_4^T\\ &-2\zeta_5^T - \cdots - \{zI_d + I_d - 2\zeta_{2n}^T - 2\zeta_{2n+1}^T\}^{-1}4\zeta_{2n}^T\zeta_{2n-1}^T\}^{-1}\cdot\\ &\cdots\cdot 4\zeta_4^T\zeta_3^T\}^{-1}4\zeta_2^T\zeta_1^T\}^{-1}S_0^{1/2}, \end{aligned} \tag{4.13}$$

and an iterative application of the matrix identity

$$I_d + A^{-1}B = (I_d - (B + A)^{-1}B)^{-1}$$

and Markov's theorem (see Duran (1996)) gives

$$\begin{aligned} \int \frac{d\Sigma(x)}{z - x} = \lim_{n\to\infty} S_0^{1/2}\Big\{&(z+1)I_d - \Big\{I_d - \Big\{(z+1)I_d -\\ &\cdots - \Big\{(z+1)I_d - 2\zeta_{2n+1}^T\Big\}^{-1}2\zeta_{2n}^T\Big\}^{-1}\ldots 2\zeta_2^T\Big\}^{-1}2\zeta_1^T\Big\}^{-1}S_0^{1/2} \end{aligned}$$

(note that this transformation is essentially a contraction). This proves the first part of the theorem. For the second part we put $z = 1$ and use formula (1.3) in Fair (1971) to obtain

$$\begin{aligned} \int \frac{d\Sigma(x)}{1 - x} &= \lim_{n\to\infty} \frac{1}{2}S_0^{1/2}\Big\{I_d - \Big\{I_d - \Big\{I_d -\\ &\qquad \cdots - \Big\{I_d - \zeta_{2n+1}^T\Big\}^{-1}\zeta_{2n}^T\Big\}^{-1}\ldots\Big\}^{-1}\zeta_1^T\Big\}^{-1}S_0^{1/2}\\ &= \lim_{n\to\infty} \frac{1}{2}S_0^{1/2}\sum_{j=0}^{n+1} X_{j+1}^{-1}\zeta_j^T X_{j-1}X_j^{-1}\zeta_{j-1}^T X_{j-2}X_{j-1}^{-1}\ldots X_1X_2^{-1}\zeta_1^T S_0^{1/2}, \end{aligned} \tag{4.14}$$

where $X_0 = I_d, X_1 = I_d$,

$$X_{n+1} = X_n - \zeta_n^T X_{n-1} \quad (n \geq 1).$$

Now a straightforward induction argument shows that $X_{n+1} = V_n^T \dots V_1^T$ and (4.14) reduces to (4.4), which proves the remaining assertion of the theorem. □

Our next result generalizes the famous characterization of recurrence in an irreducible birth-and-death chain to the matrix case.

THEOREM 4.5. [illegible] 2.1 [illegible] $[-1,1]$. [illegible] $(0,\ell)$ [illegible]

$$e_\ell^T S_0^{1/2} \sum_{i=0}^{\infty} T_{i+1}^{-1} A_i^{-1} C_i^T T_{i-1} T_i^{-1} A_{i-1}^{-1} C_{i-1}^T T_{i-2} T_{i-1}^{-1} \cdot \\ \cdots T_1 T_2^{-1} A_1^{-1} C_1^T T_0 T_1^{-1} A_0^{-1} T_0 S_0^{-1/2} e_\ell = \infty,$$

[illegible] $T_i = Q_i(1)$ $(i \in \mathbb{N}_0)$ $T_{-2} = T_{-1} = I_d$ [illegible] $Q_i(x)$ [illegible] $i$ [illegible] (1.4) [illegible] $\mathcal{C}_d$ [illegible]

$$S_0^{1/2} \sum_{i=0}^{\infty} T_{i+1}^{-1} A_i^{-1} C_i^T T_{i-1} T_i^{-1} A_{i-1}^{-1} C_{i-1}^T T_{i-2} T_{i-1}^{-1} \dots T_1 T_2^{-1} A_1^{-1} C_1^T T_0 T_1^{-1} A_0^{-1} T_0 S_0^{-1/2}$$

[illegible]

[illegible] A combination of Corollary 4.1 and Theorem 4.4 shows that the state $(0,\ell)$ is recurrent if and only if

$$(4.15) \qquad t = \frac{1}{2} e_\ell^T S_0^{1/2} \left[ I_d + \sum_{j=1}^{\infty} (V_j^T)^{-1} \dots (V_l^T)^{-1} U_j^T \dots U_1^T \right] S_0^{-1/2} e_\ell = \infty,$$

where $U_1, U_2, \dots$ are the canonical moments of the random walk measure $\Sigma$ and $V_j = I_d - U_j$ $(j \geq 1)$. In the following we express the right-hand side in terms of the blocks of the one-step block tridiagonal transition matrix $P$ corresponding to the random walk. For this consider the recurrence relation (1.4) and define $T_n = Q_n(1)$. Note that the polynomials $\underline{Q}_n(t) = A_0 \dots A_{n-1} Q_n(t)$ are monic and satisfy the recurrence relation

$$\underline{Q}_{n+1}(t) = t\underline{Q}_n(t) - A_0 \dots A_{n-1} B_n A_{n-1}^{-1} \dots A_0^{-1} \underline{Q}_n(t) \\ - A_0 \dots A_{n-1} C_n^T A_{n-2}^{-1} \dots A_0^{-1} \underline{Q}_{n-1}(t).$$

Therefore a comparison with (4.5) yields

$$(4.16) \qquad \begin{aligned} A_0 \dots A_{n-1} B_n A_{n-1}^{-1} \dots A_0^{-1} &= -I_d + 2\zeta_{2n}^T + 2\zeta_{2n+1}^T, \\ A_0 \dots A_{n-1} C_n^T A_{n-2}^{-1} \dots A_0^{-1} &= 4\zeta_{2n}^T \zeta_{2n-1}^T. \end{aligned}$$

Using these representations and the fact that $U_k V_k = V_k U_k$ (see Dette and Studden (2001), Theorem 2.7) it is easy to see that

$$T_n = Q_n(1) = 2^n A_{n-1}^{-1} \dots A_0^{-1} V_{2n-1}^T V_{2n-2}^T \dots V_1^T,$$

and it follows from the same reference that these matrices are nonsingular for all $n \in \mathbb{N}_0$. Therefore we can define

$$(4.17) \qquad \hat{Q}_n(x) = T_n^{-1} Q_n(x),$$

and it is easy to see that these polynomials satisfy the recurrence relation

$$x\hat{Q}_n(x) = \hat{A}_n\hat{Q}_{n+1}(x) + \hat{B}_n\hat{Q}_n(x) + \hat{C}_n^T\hat{Q}_{n-1}(x), \tag{4.18}$$

where

$$\hat{A}_n = T_n^{-1}A_nT_{n+1}, \quad \hat{B}_n = T_n^{-1}B_nT_n, \quad \hat{C}_n^T = T_n^{-1}C_n^TT_{n-1} \tag{4.19}$$

(note that $\hat{A}_n + \hat{B}_n + \hat{C}_n^T = I_d$). Combining (4.16) with (4.19) we obtain

$$\begin{aligned}\hat{A}_0\dots\hat{A}_{n-1}\hat{B}_n\hat{A}_{n-1}^{-1}\dots\hat{A}_0^{-1} &= -I_d + 2\zeta_{2n}^T + 2\zeta_{2n+1}^T,\\ \hat{A}_0\dots\hat{A}_{n-1}\hat{C}_n^T\hat{A}_{n-2}^{-1}\dots\hat{A}_0^{-1} &= 4\zeta_{2n}^T\zeta_{2n-1}^T,\end{aligned}$$

and by an induction argument (noting that $\hat{A}_n + \hat{B}_n + \hat{C}_n^T = I_d$) it follows that

$$\begin{aligned}2U_{2n}^TU_{2n-1}^T &= \hat{A}_0\dots\hat{A}_{n-1}\hat{C}_n^T\hat{A}_{n-1}^{-1}\dots\hat{A}_0^{-1},\\ 2V_{2n+1}^TV_{2n}^T &= \hat{A}_0\dots\hat{A}_{n-1}\hat{A}_n\hat{A}_{n-1}^{-1}\dots\hat{A}_0^{-1}.\end{aligned}$$

Finally, we obtain for the left-hand side of (4.15)

$$\begin{aligned}t &= \frac{1}{2}e_\ell^TS_0^{1/2}\sum_{j=0}^{\infty}\Big\{(V_1^T)^{-1}\dots(V_{2j}^T)^{-1}U_{2j}^T\dots U_1^T\\ &\qquad\qquad + (V_1^T)^{-1}\dots(V_{2j+1}^T)^{-1}U_{2j+1}^T\dots U_1^T\Big\}S_0^{-1/2}e_\ell\\ &= e_\ell^TS_0^{1/2}\sum_{j=0}^{\infty}\hat{A}_j^{-1}\hat{C}_j^T\hat{A}_{j-1}^{-1}\dots\hat{A}_1^{-1}\hat{C}_1^T\hat{A}_0^{-1}S_0^{-1/2}e_\ell\\ &= e_\ell^TS_0^{1/2}\sum_{j=0}^{\infty}T_{j+1}^{-1}A_j^{-1}C_j^TT_{j-1}T_j^{-1}A_{j-1}^{-1}C_{j-1}^TT_{j-2}T_{j-1}^{-1}\cdot\\ &\qquad\qquad \cdots T_1T_2^{-1}A_1^{-1}C_1^TT_0T_1^{-1}A_0^{-1}T_0S_0^{-1/2}e_\ell\end{aligned}$$

with $T_i = Q_i(1)$ $(i \in \mathbb{N}_0)$, which proves the assertion of the theorem. □

4.6. It is interesting to note that the condition in Theorem 4.5 simplifies substantially if all the matrices $T_i, A_i, C_i$ are commuting. In this case an irreducible random walk is recurrent if and only if

$$e_\ell^TS_0^{1/2}\sum_{i=0}^{\infty}T_{i+1}^{-1}T_i^{-1}(C_1\dots C_i)^T(A_0\dots A_i)^{-1}S_0^{-1/2}e_\ell = \infty$$

for some $\ell \in \{1,\dots,d\}$.

4.7. Consider the random walk on the tree introduced in section 3.5. By Corollary 4.1 the state $(0,1)$ (which corresponds to the origin) is recurrent if and only if

$$\infty = e_1^T\left(\int\frac{d\Sigma(x)}{1-x}\right)\left(\int d\Sigma(x)\right)^{-1}e_1 = \int_{2\sqrt{pq}}^{2\sqrt{pq}}\frac{a(x)}{1-x}dx,$$

where the function $a$ is defined in section 3.5 and we have used the fact that $\int d\Sigma(x) = S_0 = (R_0^TR_0)^{-1}$ (see Remark 2.2). Because the support of the spectral measure is given by the interval $[-\sqrt{4pq}, -\sqrt{4pq}]$ it follows that the condition $p = q = \frac{1}{2}$ is necessary for the recurrence of the random walk. Now a straightforward calculation shows that the state $(0,1)$ (i.e., the center of the graph) is recurrent if and only if the condition $2\sum_{i=2}^d d_i^2 = \sum_{i=2}^d d_i$ is satisfied (in all other cases the integral is finite).

## 5. Applications.

In this section we briefly discuss some applications of our approach.

### 5.1. Representations of the invariant measure.

Note that an irreducible quasi-birth-and-death process always has an invariant measure with a matrix product form (see Latouche, Pearce, and Taylor (1998)). In particular, if the process is positive recurrent, the invariant measure coincides with the stationary distribution $x = (x_0^T, x_1^T, \dots)$, which can be represented as

$$x_k^T = x_0^T \prod_{\ell=0}^{k-1} \tilde{R}_\ell, \tag{5.1}$$

where the set $\{\tilde{R}_\ell\}_{\ell=0}^\infty$ is the minimal nonnegative solution of the equations

$$R_k = A_k + R_k B_{k+1} + R_k R_{k+1} C_{k+2}^T \quad (k \geq 0) \tag{5.2}$$

and $x_0$ satisfies

$$x_0^T (B_0 + \tilde{R}_0 C_1^T) = x_0^T, \tag{5.3}$$

normalized so that $x^T e = 1$, where $e$ denotes a vector with all entries equal to one. We will now investigate these properties from the viewpoint derived in this paper and suppose that the assumptions of Theorem 2.5 are satisfied for an irreducible aperiodic Markov chain on the grid $\mathcal{C}_d$. Note that the limits

$$L_{i'} = \lim_{n\to\infty} P_{ii'}^n$$

exist and do not depend on $i$. By the Theorem of dominated convergence and (2.16) it follows that

$$L_{i'} = \lim_{n\to\infty} \left\{ Q_i(1)\Sigma(1)Q_{i'}^T(1) + (-1)^n Q_i(-1)\Sigma(-1)Q_{i'}^T(-1) \right\} Z_{i'}^{-1}, \tag{5.4}$$

where $Z_{i'} = \int Q_{i'}(x) d\Sigma(x) Q_{i'}^T(x)$ and $\Sigma(1)$ and $\Sigma(-1)$ denote the mass of the random walk matrix measure at the points 1 and $-1$, respectively. Considering the subsequence of odd positive integers it follows that $\Sigma$ has no mass at $-1$ and (5.4) reduces to

$$L_{i'} = \lim_{n\to\infty} P_{ii'}^n = Q_i(1)\Sigma(1)Q_{i'}^T(1) Z_{i'}^{-1}. \tag{5.5}$$

Note that the left-hand side of this equation does not depend on $i$ and therefore (5.5) provides several representations for the same quantity (by using different values of $i$). For example, if we put $i = 0$ and note that the rank of the matrices $L_{i'}$ is 1 we obtain from the identity $L_0 = \Sigma(1) Z_0^{-1}$ that the rank of the weight $\Sigma(1)$ is 1. Moreover, if the random walk is positive recurrent, the stationary distribution is given by

$$x = (x_0^T, x_1^T, \dots) = e_0^T (L_0, L_1, \dots),$$

and it follows that

$$x_k^T = e_0^T L_k = e_0^T \Sigma(1) Q_k^T(1) Z_k^{-1} \quad (k \geq 0),$$

which is an alternative representation for the stationary distribution. In particular, we have for the vector $x_0$ in (5.1)

$$x_0^T = e_0^T \Sigma(1) S_0^{-1}.$$

Moreover, if the matrices $Q_k(1)$ are nonsingular, we obtain by straightforward calculation that the representation (5.1) holds with

$$\tilde{R}_j = Z_j (Q_j^T(1))^{-1} Q_{j+1}^T(1) Z_{j+1}^{-1}. \tag{5.6}$$

Using the relations (2.4) it follows by straightforward algebra that the sequence $\{\tilde{R}_j\}_{j\in\mathbb{N}_0}$ is in fact a solution of the system (5.2) and (5.3), which yield to the stationary distribution. We finally note that the proof of Theorem 2.1 shows that the matrix $Z_j^{-1} = R_j^T R_j$ can be expressed in terms of the blocks $A_j, C_j$ and the matrix $S_0$.

**5.2. A necessary condition for positive recurrence.** As a second application we use the identity (5.5) for two values $i, k$ and obtain

$$Q_i(1)\Sigma(1)Q_{i'}^T(1) = Q_k(1)\Sigma(1)Q_{i'}^T(1) = L_{i'}$$

which reduces for $i' = 0$ to

$$(Q_i(1) - Q_k(1))\Sigma(1) = 0 \quad (i, k \geq 0). \tag{5.7}$$

Recall that by Corollary 4.2 the irreducible random walk is positive recurrent if and only if all measures $e_\ell^T d\Sigma(x) S_0^{-1} e_\ell$ have a jump at the point 1. In this case it follows from (5.7) that all matrices $Q_i(1) - Q_k(1)$ are singular (otherwise $\Sigma(1)$ would be the null matrix). Consequently we obtain the following result.

THEOREM 5.1. [illegible] 2.5 [illegible]

$$Q_i(1) - Q_k(1)$$

[illegible] $i, k \in \mathbb{N}_0$.

[illegible] 5.2. Consider the random walk on the tree presented in section 3.5. In Example 4.7 it is demonstrated that the random walk is recurrent if and only if $p = q = \frac{1}{2}$ and

$$\sum_{i=2}^{d} d_i = 2\sum_{i=2}^{d} d_i^2,$$

which will be assumed in the following discussion. A straightforward calculation shows that $Q_0(1) = I_d$,

$$Q_1(1) = \begin{pmatrix} 2 & -2d_2 & -2d_3 & \dots & -2d_d \\ -1 & 2 & 0 & \dots & 0 \\ -1 & 0 & 2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \dots & 2 \end{pmatrix},$$

and consequently we obtain

$$|Q_1(1) - Q_0(1)| = |I_d - 2B_0| = 1 - 2d_2 - \cdots - 2d_d = 2d_1 - 1.$$

Therefore, if the random walk would be positive recurrent it follows that $d_1 = \frac{1}{2}$. Because the tree corresponding to the random walk is symmetric we conclude that the role of $d_1$ and $d_2$ can be interchanged. Consequently the random walk can only be positive recurrent if two of the probabilities $d_j$ are equal to $1/2$ and the others vanish. However, this corresponds to the symmetric random walk on $\mathbb{Z}$, which is not positive recurrent. In other words: the random walk considered in section 3.5 is recurrent if $p = q = \frac{1}{2}$ and $\sum_{i=2}^{d} d_i = 2\sum_{i=2}^{d} d_i^2$ but never positive recurrent.

**Acknowledgments.** This work was done while H. Dette was visiting the Department of Statistics, Purdue University, and this author would like to thank the Department for its hospitality. The authors would like to thank two unknown referees for their constructive comments on an earlier version of this paper and Isolde Gottschlich, who typed this paper with considerable technical expertise.

## REFERENCES

A. Basilevsky (1983), *Applied Matrix Algebra in the Statistical Sciences*, North–Holland, Amsterdam.

N. G. Bean, P. K. Pollett, and P. G. Taylor (2000), *Quasi-stationary distributions for level-dependent quasi-birth-and-death processes*, Comm. Statist. Stochastic Models, 16, pp. 511–541.

Ju. M. Berezanskii (1968), *Expansions in Eigenfunctions of Selfadjoint Operators*, Trans. Math. Monogr. 17, AMS, Providence, RI.

L. Bright and P. G. Taylor (1995), *Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes*, Comm. Statist. Stochastic Models, 11, pp. 497–525.

T. Dayar and F. Quessette (2002), *Quasi-birth-and-death processes with level-geometric distribution*, SIAM J. Matrix Anal. Appl., 24, pp. 281–291.

H. Dette (1996), *On the generating functions of a random walk on the nonnegative integers*, J. Appl. Probab., 33, pp. 1033–1052.

H. Dette and W. J. Studden (1997), *The Theory of Canonical Moments with Applications in Statistics, Probability, and Analysis*, Wiley, New York.

H. Dette and W. J. Studden (2001), *Matrix measures, moment spaces, and Favard's theorem for the interval* $[0,1]$ *and* $[0,\infty)$, Linear Algebra Appl., 345, pp. 163–193.

A. J. Duran (1995), *On orthogonal polynomials with respect to a positive definite matrix of measures*, Canad. J. Math., 47, pp. 88–112.

A. J. Duran (1996), *Markov's theorem for orthogonal matrix polynomials*, Canad. J. Math., 48, pp. 1180–1195.

A. J. Duran (1999), *Ratio asymptotics for orthogonal matrix polynomials*, J. Approx. Theory, 100, pp. 304–344.

A. J. Duran and W. Van Assche (1995), *Orthogonal matrix polynomials and higher-order recurrence relations*, Linear Algebra Appl., 219, pp. 261–280.

W. Fair (1971), *Noncommutative continued fractions*, SIAM J. Math. Anal., 2, pp. 226–232.

W. Feller (1950), *An Introduction to Probability Theory and Its Applications, Vol.* I, John Wiley & Sons, New York.

D. P. Gaver, P. A. Jacobs, and G. Latouche (1984), *Finite birth-and-death models in randomly changing environments*, Adv. in Appl. Probab., 16, pp. 715–731.

B. Hajek (1982), *Birth-and-death processes on the integers with phases and general boundaries*, J. Appl. Probab., 19, pp. 488–499.

P. R. Halmos and V. S. Sunder (1978), *Bounded Integral Operators on* $L^2$*-Spaces*, Springer-Verlag, New York.

S. Karlin and J. McGregor (1959), *Random walks*, Illionis J. Math., 3, pp. 66–81.

S. Karlin and H. M. Taylor (1975), *A First Course in Stochastic Processes*, Academic Press, New York.

G. Latouche, C. E. M. Pearce, and P. G. Taylor (1998), *Invariant measures for quasi-birth-and-death processes*, Comm. Statist. Stochastic Models, 14, pp. 443–460.

G. Latouche and V. Ramaswami (1999), *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ASA-SIAM Ser. Stat. Appl. Probab. 5, SIAM, Philadelphia, Chapter 12.

Q. L. Li and J. Cao (2004), *Two types of RG-factorizations of quasi-birth-and-death processes and their applications to stochastic integral functionals*, Stoch. Models, 20, pp. 299–340.

F. Marcellán and G. Sansigre (1993), *On a class of matrix orthogonal polynomials on the real line*, Linear Algebra Appl., 181, pp. 97–109.

I. Marek (2003), *Quasi-birth-and-death processes, level geometric distributions. An aggregation/disaggregation approach*, J. Comput. Appl. Math., 152, pp. 277–288.

M. F. Neuts (1981), *Matrix Geometric Solutions in Stochastic Models. An Algorithmic Approach.* The John Hopkins University Press, Baltimore.

M. F. Neuts (1989), *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, Marcel Dekker, New York.

A. Ost (2001), *Performance of Communication Systems: A Model-Based Approach with Matrix-Geometric Methods*, Springer-Verlag, Berlin.

V. Ramaswami and P. G. Taylor (1996), *Some properties of the rate operators in level dependent quasi-birth-and-death processes with a countable number of phases*, Comm. Statist. Stochastic Models, 12, pp. 143–164.

L. Rodman (1990), *Orthogonal matrix polynomials*, in Orthogonal Polynomials, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci. 294, P. Nevai, ed., Kluwer, Dordrecht, The Netherlands, pp. 345–362.

A. Sinap and W. Van Assche (1996), *Orthogonal matrix polynomials and applications*, J. Comput. Appl. Math., 66, pp. 27–52.

E. A. van Doorn and P. Schrijner (1993), *Random walk polynomials and random walk measures*, J. Comput. Appl. Math., 49, pp. 289–296.

E. A. van Doorn and P. Schrijner (1995), *Geometric ergodicity and quasi-stationarity in discrete-time birth-death processes*, J. Austral. Math. Soc. Ser. B, 37, pp. 121–144.

T. A. Whitehurst (1982), *An application of orthogonal polynomials to random walks*, Pacific J. Math., 99, pp. 205–213.

W. Woess (1985), *Random walks and periodic continued fractions*, Adv. in Appl. Probab., 17, pp. 67–84.

M. J. Zygmunt (2002), *Matrix Chebyshev polynomials and continued fractions*, Linear Algebra Appl., 340, pp. 155–168.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 1, pp. 143–159

# SYMMETRIC LINEARIZATIONS FOR MATRIX POLYNOMIALS*

NICHOLAS J. HIGHAM[†], D. STEVEN MACKEY[†], NILOUFER MACKEY[‡], AND FRANÇOISE TISSEUR[†]

**Abstract.** A standard way of treating the polynomial eigenvalue problem $P(\lambda)x = 0$ is to convert it into an equivalent matrix pencil—a process known as linearization. Two vector spaces of pencils $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$, and their intersection $\mathbb{DL}(P)$, have recently been defined and studied by Mackey, Mackey, Mehl, and Mehrmann. The aim of our work is to gain new insight into these spaces and the extent to which their constituent pencils inherit structure from $P$. For arbitrary polynomials we show that every pencil in $\mathbb{DL}(P)$ is block symmetric and we obtain a convenient basis for $\mathbb{DL}(P)$ built from block Hankel matrices. This basis is then exploited to prove that the first $\deg(P)$ pencils in a sequence constructed by Lancaster in the 1960s generate $\mathbb{DL}(P)$. When $P$ is symmetric, we show that the symmetric pencils in $\mathbb{L}_1(P)$ comprise $\mathbb{DL}(P)$, while for Hermitian $P$ the Hermitian pencils in $\mathbb{L}_1(P)$ form a proper subset of $\mathbb{DL}(P)$ that we explicitly characterize. Almost all pencils in each of these subsets are shown to be linearizations. In addition to obtaining new results, this work provides a self-contained treatment of some of the key properties of $\mathbb{DL}(P)$ together with some new, more concise proofs.

**Key words.** matrix polynomial, matrix pencil, linearization, companion form, quadratic eigenvalue problem, vector space, block symmetry, Hermitian, Hankel

**AMS subject classifications.** 65F15, 15A18

**DOI.** 10.1137/050646202

## 1. Introduction.

The polynomial eigenvalue problem $P(\lambda)x = 0$, where

$$P(\lambda) = \sum_{i=0}^{k} \lambda^i A_i, \qquad A_i \in \mathbb{C}^{n\times n}, \quad A_k \neq 0, \tag{1.1}$$

arises in many applications and is an active topic of study. The quadratic case ($k = 2$) is the most important in practice [25], but higher degree polynomials also arise [5], [13], [19], [24]. We continue the practice stemming from Lancaster [15] of developing theory for general $k$ where possible, in order to gain the most insight and understanding.

The standard way of solving the polynomial eigenvalue problem is to linearize $P(\lambda)$ into $L(\lambda) = \lambda X + Y \in \mathbb{C}^{kn\times kn}$, solve the generalized eigenproblem $L(\lambda)z = 0$, and recover eigenvectors of $P$ from those of $L$. Formally, $L$ is a linearization of $P$ if there exist unimodular $E(\lambda)$ and $F(\lambda)$ (that is, $\det(E(\lambda))$ and $\det(F(\lambda))$ are nonzero constants) such that

$$E(\lambda)L(\lambda)F(\lambda) = \begin{bmatrix} P(\lambda) & 0 \\ 0 & I_{(k-1)n} \end{bmatrix}.$$

Hence $\det(L(\lambda))$ agrees with $\det(P(\lambda))$ up to a nonzero constant multiplier, so that $L$ and $P$ have the same eigenvalues. The linearizations used in practice are almost

---

[*] Received by the editors November 28, 2005; accepted for publication (in revised form) by Z. Bai July 3, 2006; published electronically December 21, 2006. This work was supported by Engineering and Physical Sciences Research Council grant GR/S31693.

http://www.siam.org/journals/simax/29-1/64620.html

[†] School of Mathematics, The University of Manchester, Sackville Street, Manchester, M60 1QD, UK (higham@ma.man.ac.uk, smackey@ma.man.ac.uk, ftisseur@ma.man.ac.uk). The work of the first author was supported by a Royal Society-Wolfson Research Merit Award.

[‡] Department of Mathematics, Western Michigan University, Kalamazoo, MI 49008 (nil.mackey@wmich.edu).

invariably one of $C_1(\lambda) = \lambda X_1 + Y_1$ and $C_2(\lambda) = \lambda X_2 + Y_2$, called the first and second companion forms [16, sect. 14.1], respectively, where

$$X_1 = X_2 = \mathrm{diag}(A_k, I_n, \ldots, I_n), \tag{1.2a}$$

$$Y_1 = \begin{bmatrix} A_{k-1} & A_{k-2} & \ldots & A_0 \\ -I_n & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & -I_n & 0 \end{bmatrix}, \quad Y_2 = \begin{bmatrix} A_{k-1} & -I_n & \ldots & 0 \\ A_{k-2} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -I_n \\ A_0 & 0 & \ldots & 0 \end{bmatrix}. \tag{1.2b}$$

Yet many linearizations exist, and other than the convenience of their construction, there is no apparent reason for preferring the companion forms. Indeed one obvious disadvantage of the companion forms is their lack of preservation of certain structural properties of $P$, most obviously symmetry.

Four recent papers have systematically addressed the task of broadening the menu of available linearizations and providing criteria to guide the choice. Mackey et al. [17] construct two vector spaces of pencils generalizing the companion forms and prove many interesting properties, including that almost all of these pencils are linearizations. In [18], the same authors identify linearizations within these vector spaces that respect palindromic and odd-even structures. Higham, D. S. Mackey, and Tisseur [10] analyze the conditioning of some of the linearizations introduced in [17], looking for a best conditioned linearization and comparing its condition number with that of the original polynomial. Most recently, Higham, Li, and Tisseur [9] investigate the backward error of approximate eigenpairs recovered from a linearization, obtaining results complementary to, but entirely consistent with, those of [10].

Before discussing our aims, we recall some definitions and results from [17]. Let $\mathbb{F}$ denote $\mathbb{C}$ or $\mathbb{R}$. With the notation

$$\Lambda = [\lambda^{k-1}, \lambda^{k-2}, \ldots, 1]^T \in \mathbb{F}^k, \quad \text{where} \quad k = \deg(P),$$

define two vector spaces of $kn \times kn$ pencils $L(\lambda) = \lambda X + Y$:

$$\mathbb{L}_1(P) = \left\{ L(\lambda) : L(\lambda)(\Lambda \otimes I_n) = v \otimes P(\lambda),\ v \in \mathbb{F}^k \right\}, \tag{1.3}$$

$$\mathbb{L}_2(P) = \left\{ L(\lambda) : (\Lambda^T \otimes I_n)L(\lambda) = w^T \otimes P(\lambda),\ w \in \mathbb{F}^k \right\}. \tag{1.4}$$

The vectors $v$ and $w$ are referred to as "right ansatz" and "left ansatz" vectors, respectively. It is easily checked that for the companion forms in (1.2), $C_1(\lambda) \in \mathbb{L}_1(P)$ with $v = e_1$ and $C_2(\lambda) \in \mathbb{L}_2(P)$ with $w = e_1$, where $e_i$ denotes the $i$th column of $I_k$. The dimensions of $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$ are both $k(k-1)n^2 + k$ [17, Cor. 3.6]. For any regular $P$ (that is, any $P$ for which $\det(P(\lambda)) \not\equiv 0$), almost all pencils in $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$ are linearizations of $P$ [17, Thm. 4.7].

A crucial property of $\mathbb{L}_1$ and $\mathbb{L}_2$ is that eigenvectors of $P$ can be directly recovered from eigenvectors of linearizations in $\mathbb{L}_1$ and $\mathbb{L}_2$. Specifically, for any pencil $L \in \mathbb{L}_1(P)$ with nonzero right ansatz vector $v$, $x$ is a right eigenvector of $P$ with eigenvalue $\lambda$ if and only if $\Lambda \otimes x$ (if $\lambda$ is finite) or $e_1 \otimes x$ (if $\lambda = \infty$) is a right eigenvector for $L$ with eigenvalue $\lambda$. Moreover, if this $L \in \mathbb{L}_1(P)$ is a linearization for $P$, then right eigenvector of $L$ has one of these two Kronecker product forms; hence some right eigenvector of $P$ can be recovered from every right eigenvector of $L$. A similar recovery property holds for left eigenvectors and pencils in $\mathbb{L}_2(P)$. For more details, see [17, Thms. 3.8, 3.14, and 4.4].

The subspace

$$\mathbb{DL}(P) = \mathbb{L}_1(P) \cap \mathbb{L}_2(P) \tag{1.5}$$

of "double ansatz" pencils is of particular interest, because there is a simultaneous correspondence via Kronecker products between left [illegible] right eigenvectors of $P$ and those of pencils in $\mathbb{DL}(P)$. Two key facts are that $L \in \mathbb{DL}(P)$ if and only if $L$ satisfies the conditions in (1.3) and (1.4) with $w = v$, and that every $v \in \mathbb{F}^k$ uniquely determines $X$ and $Y$ such that $L(\lambda) = \lambda X + Y$ is in $\mathbb{DL}(P)$ [17, Thm. 5.3]. Thus $\mathbb{DL}(P)$ is a $k$-dimensional space of pencils associated with $P$. Just as for $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$, almost all pencils in $\mathbb{DL}(P)$ are linearizations [17, Thm. 6.8].

Our contributions are now summarized. We show in section 3 that the set of block symmetric pencils in $\mathbb{L}_1(P)$ is precisely $\mathbb{DL}(P)$. That $\mathbb{DL}(P)$ should comprise only block symmetric pencils is perhaps surprising, as $P$ is arbitrary. We show that the pencils corresponding to $v = e_i$, $i = 1\colon k$, form a basis for $\mathbb{DL}(P)$ built from block diagonal matrices with block Hankel blocks. This basis is used in section 4 to prove that the first $k = \deg(P)$ pencils in a sequence constructed by Lancaster [14], [15], generate $\mathbb{DL}(P)$. In sections 5 and 6 we show that when $P$ is symmetric the set of symmetric pencils in $\mathbb{L}_1(P)$ is the same as $\mathbb{DL}(P)$, while for Hermitian $P$ the Hermitian pencils in $\mathbb{L}_1(P)$ form a proper subset of $\mathbb{DL}(P)$ corresponding to real ansatz vectors. In section 7 we summarize the known "almost all pencils are linearizations" results and prove such a result for the Hermitian pencils in $\mathbb{L}_1(P)$.

Initially, our main motivation for this investigation was the problem of systematically generating symmetric linearizations for symmetric matrix polynomials. However, the analysis has led, via the study of [illegible] pencils, to new derivations of some of the general properties of $\mathbb{DL}(P)$. Therefore this paper should be useful as a self-contained introduction to $\mathbb{DL}(P)$ with proofs that are conceptually clearer and more concise than the original derivations in [17].

Finally, we motivate our interest in the preservation of symmetry. A matrix polynomial that is real symmetric or Hermitian has a spectrum that is symmetric with respect to the real axis, and the sets of left and right eigenvectors coincide. These properties are preserved in a symmetric (Hermitian) linearization by virtue of its structure—not just through the numerical entries of the pencil. A symmetry-preserving pencil has the practical advantages that storage and computational cost are reduced if a method that exploits symmetry is applied. The eigenvalues of a symmetric (Hermitian) pencil $L(\lambda) = \lambda X + Y$ can be computed, for small to medium size problems, by first reducing the matrix pair $(Y, X)$ to tridiagonal-diagonal form [23] and then using the HR [4], [6] or LR [21] algorithms or the Ehrlich–Aberth iterations [3]. For large problems, a symmetry-preserving pseudo-Lanczos algorithm of Parlett and Chen [20], [2, sect. 8.6], based on an indefinite inner product, can be used. For a quadratic polynomial $Q(\lambda)$ that is hyperbolic, or in particular overdamped, a linearization that is a symmetric definite pencil can be identified [11, Thm. 3.6]; this pencil is amenable to structure-preserving methods that exploit both the symmetry and the definiteness [27] and guarantee real computed eigenvalues for $Q(\lambda)$ not too close to being nonhyperbolic.

**2. Block symmetry and shifted sum.** We begin with some notation and results concerning block transpose and block symmetry. Our aim is to investigate the existence and uniqueness of solutions in block symmetric matrices of the equation $X \boxplus\!\!\rightarrow Y = Z$, where $\boxplus\!\!\rightarrow$ is a "shifted sum" operation and $Z$ is a given arbitrary matrix. For the purposes of this paper we consider only block matrices in which all the blocks have the same size.

Definition 2.1 (block transpose). [illegible] $A = (A_{ij})$ [illegible] $k \times \ell$ [illegible] $m \times n$ [illegible] $A_{ij}$ [illegible] $A$ [illegible] $\ell \times k$ [illegible] $A^{\mathcal{B}}$ [illegible] $m \times n$ [illegible] $(A^{\mathcal{B}})_{ij} = A_{ji}$

Recall that all pencils in $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$ are of size $kn \times kn$, where $k$ is the degree of the $n \times n$ matrix polynomial $P(\lambda)$. Throughout this paper we regard these pencils as block $k \times k$ matrices with $n \times n$ blocks. The block transpose operation, performed relative to this partitioning, establishes an intimate link between $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$.

Theorem 2.2. [illegible] $P(\lambda)$ [illegible]

$$\mathbb{L}_1(P) \longrightarrow \mathbb{L}_2(P),$$

$$L(\lambda) \longmapsto L(\lambda)^{\mathcal{B}}$$

[illegible] $\mathbb{L}_1(P)$ [illegible] $\mathbb{L}_2(P)$ [illegible] $L(\lambda) \in \mathbb{L}_1(P)$ [illegible] $v$ [illegible] $L(\lambda)^{\mathcal{B}} \in \mathbb{L}_2(P)$ [illegible] $w = v$

[illegible] It is straightforward to check that $\big(L(\lambda)(\Lambda \otimes I_n)\big)^{\mathcal{B}} = (\Lambda \otimes I_n)^{\mathcal{B}} L(\lambda)^{\mathcal{B}} = (\Lambda^T \otimes I_n) L(\lambda)^{\mathcal{B}}$ and $\big(v \otimes P(\lambda)\big)^{\mathcal{B}} = v^T \otimes P(\lambda)$. Hence if $L(\lambda) \in \mathbb{L}_1(P)$ with right ansatz vector $v$, then block transposing the defining condition in (1.3) yields $(\Lambda^T \otimes I_n) L(\lambda)^{\mathcal{B}} = v^T \otimes P(\lambda)$. Thus $L(\lambda)^{\mathcal{B}} \in \mathbb{L}_2(P)$ with left ansatz vector $v$, and so block transpose gives a well-defined map from $\mathbb{L}_1(P)$ to $\mathbb{L}_2(P)$. Clearly this map is linear and the kernel is just the zero pencil, since $L(\lambda)^{\mathcal{B}} = 0 \Rightarrow L(\lambda) = 0$. Since $\dim \mathbb{L}_1(P) = \dim \mathbb{L}_2(P)$, the proof is complete. ☐

The companion forms give a nice illustration of Theorem 2.2. By inspection, $C_2(\lambda) = \big(C_1(\lambda)\big)^{\mathcal{B}}$ and, as noted earlier, $C_1(\lambda) \in \mathbb{L}_1(P)$ with right ansatz vector $v = e_1$ while $C_2(\lambda) \in \mathbb{L}_2(P)$ with left ansatz vector $w = v = e_1$.

Given the notion of block transpose, it is natural to consider block symmetric matrices, which will play a central role in our development. A block $k \times k$ matrix $A$ with $m \times n$ blocks is [illegible] if $A^{\mathcal{B}} = A$. For example, a block $2 \times 2$ block symmetric matrix has the form $\left[\begin{smallmatrix} A_{11} & A_{12} \\ A_{12} & A_{22} \end{smallmatrix}\right]$. Note that if each block $A_{ij} \in \mathbb{F}^{n \times n}$ in a block symmetric matrix $A$ is symmetric, then $A$ is symmetric.

The [illegible] introduced in [17] is a simple operation on block matrices that enables us both to easily construct pencils in $\mathbb{L}_1(P)$ and to conveniently test when a given pencil is in $\mathbb{L}_1(P)$.

Definition 2.3 (column-shifted sum). [illegible] $X$ [illegible] $Y$ [illegible] $k \times k$ [illegible] $n \times n$ [illegible] $X_{ij}$ [illegible] $Y_{ij}$ [illegible] $X \boxplus\!\!\rightarrow Y$ [illegible] $X$ [illegible] $Y$ [illegible]

$$X \boxplus\!\!\rightarrow Y := \begin{bmatrix} X_{11} & \dots & X_{1k} & 0 \\ & & & \\ X_{k1} & \dots & X_{kk} & 0 \end{bmatrix} + \begin{bmatrix} 0 & Y_{11} & \dots & Y_{1k} \\ & & & \\ 0 & Y_{k1} & \dots & Y_{kk} \end{bmatrix} \in \mathbb{F}^{kn \times k(n+1)},$$

[illegible] $n \times n$

The significance of this shifted sum operation is revealed by the following result [17, Lem. 3.4], which shows how membership in $\mathbb{L}_1(P)$ is equivalent to a specific Kronecker product form in the shifted sum.

LEMMA 2.4. $P(\lambda) = \sum_{i=0}^{k} \lambda^i A_i$ $n \times n$ $k$ $L(\lambda) = \lambda X + Y$ $kn \times kn$ $v \in \mathbb{F}^k$

$$L(\lambda) \in \mathbb{L}_1(P) \quad v \iff X \boxplus\!\!\rightarrow Y = v \otimes [A_k \; A_{k-1} \; \dots \; A_0].$$

We now show that the equation $X \boxplus\!\!\rightarrow Y = Z$ with an arbitrary $Z$ may always be solved with block symmetric $X$ and $Y$. To this end we introduce the mapping

$$(X, Y) \overset{\mathcal{S}}{\longmapsto} X \boxplus\!\!\rightarrow Y$$

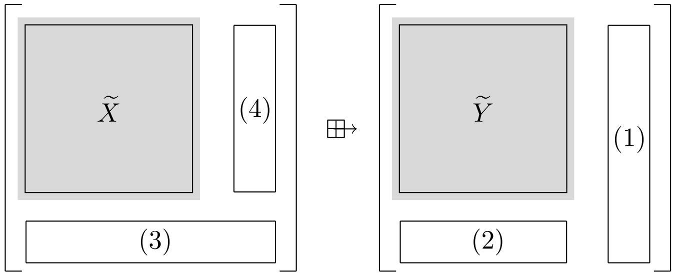
between the space of all pairs of block symmetric block $k \times k$ matrices $(X, Y)$ and the space of all block $k \times (k+1)$ matrices $Z$.

LEMMA 2.5. $\mathcal{S}$

The linearity of $\mathcal{S}$ follows easily from the definition of column-shifted sum in Definition 2.3. Since $\mathcal{S}$ is linear and its domain and codomain have the same dimension $n^2k(k+1)$, it suffices to establish either the injectivity or the surjectivity of $\mathcal{S}$; we show here the injectivity.

Suppose $X$ and $Y$ are block symmetric and $X \boxplus\!\!\rightarrow Y = 0$. We will show by induction that $X = Y = 0$. If $k = 1$ then $X \boxplus\!\!\rightarrow Y = [X \; Y]$ and $X = Y = 0$ is immediate. Let $k \geq 2$ and let $X$ and $Y$ be partitioned as indicated in the following diagram, with the last block row and last block column labeled:

$X \boxplus\!\!\rightarrow Y =$



The last block column of $X \boxplus\!\!\rightarrow Y = 0$ implies that (1) is zero. The block symmetry of $Y$ then implies that (2) is zero. The last block row of $X \boxplus\!\!\rightarrow Y = 0$ now shows that (3) is zero, and (4) is then zero by the block symmetry of $X$. We now have that

$$X \boxplus\!\!\rightarrow Y = \begin{bmatrix} \widetilde{X} & 0 \\ 0 & 0 \end{bmatrix} \boxplus\!\!\rightarrow \begin{bmatrix} \widetilde{Y} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \widetilde{X} \boxplus\!\!\rightarrow \widetilde{Y} & 0 \\ 0 & 0 \end{bmatrix} = 0.$$

Since $\widetilde{X} \boxplus\!\!\rightarrow \widetilde{Y} = 0$, the inductive hypothesis implies $\widetilde{X} = \widetilde{Y} = 0$, and consequently that $X = Y = 0$. ☐

Although Lemma 2.5 implies that $X \boxplus\!\!\rightarrow Y = Z$ can always be solved with block symmetric $X$ and $Y$, it gives no information about the form of the solution or how to construct it. Knowing the structure of $X$ and $Y$ is crucial to the later developments in sections 3 and 4, so we close this section with a procedure for explicitly constructing these solutions.

First we define three special types of block symmetric matrix that play a central role in the construction. Let

$$R_\ell = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}_{\ell\times\ell} \quad \text{and} \quad N_\ell = \begin{bmatrix} 0 & 1 & & \\ & 0 & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{bmatrix}_{\ell\times\ell}. \quad \text{(Note that } N_1 = 0.\text{)} \tag{2.1}$$

For an arbitrary $n\times n$ block $M$, we define three block Hankel, block symmetric, block $\ell\times\ell$ matrices:

$$\mathcal{H}_\ell^{(0)}(M) := \begin{bmatrix} & & M \\ & \iddots & \\ M & & \end{bmatrix} = R_\ell \otimes M,$$

$$\mathcal{H}_\ell^{(1)}(M) := \begin{bmatrix} & & M & 0 \\ & \iddots & \iddots & \\ M & \iddots & & \\ 0 & & & \end{bmatrix} = (N_\ell R_\ell)\otimes M = \begin{bmatrix} & & 1 & 0 \\ & \iddots & \iddots & \\ 1 & \iddots & & \\ 0 & & & \end{bmatrix} \otimes M,$$

$$\mathcal{H}_\ell^{(-1)}(M) := \begin{bmatrix} & & & 0 \\ & & \iddots & M \\ & \iddots & \iddots & \\ 0 & M & & \end{bmatrix} = (R_\ell N_\ell)\otimes M = \begin{bmatrix} & & & 0 \\ & & \iddots & 1 \\ & \iddots & \iddots & \\ 0 & 1 & & \end{bmatrix} \otimes M.$$

The superscript (0), (1), or (−1) denotes that the blocks $M$ are on, above, or below the antidiagonal, respectively. Note that all three of these block Hankel matrices are symmetric if $M$ is.

Now let $E_{ij}^\ell \in \mathbb{F}^{\ell\times(\ell+1)}$ denote the matrix that is everywhere zero except for a 1 in the $(i,j)$ entry. Our construction is based on the observation that for arbitrary $M, P \in \mathbb{F}^{n\times n}$, the shifted sums

$$\mathcal{H}_\ell^{(0)}(M) \boxplus\!\!\rightarrow (-\mathcal{H}_\ell^{(1)}(M)) = \begin{bmatrix} 0 & \dots & \dots & 0 & 0 \\ \vdots & & \iddots & \iddots & \vdots \\ 0 & 0 & \iddots & & \vdots \\ M & 0 & \dots & \dots & 0 \end{bmatrix} = E_{\ell 1}^\ell \otimes M, \tag{2.2}$$

$$-\mathcal{H}_\ell^{(-1)}(P) \boxplus\!\!\rightarrow \mathcal{H}_\ell^{(0)}(P) = \begin{bmatrix} 0 & \dots & \dots & 0 & P \\ \vdots & & \iddots & 0 & 0 \\ \vdots & \iddots & \iddots & & \vdots \\ 0 & 0 & \dots & \dots & 0 \end{bmatrix} = E_{1,\ell+1}^\ell \otimes P \tag{2.3}$$

place $M$ and $P$ at the bottom left corner and top right corner of a block $\ell\times(\ell+1)$ matrix, respectively.

Inherent in the linearity of the map $\mathcal{S}$ is the compatibility of the shifted sum $\boxplus\!\!\rightarrow$ with ordinary sums, i.e.,

$$\Big(\sum X_i\Big) \boxplus\!\!\rightarrow \Big(\sum Y_i\Big) = \sum (X_i \boxplus\!\!\rightarrow Y_i).$$

Hence if we can show how to construct block symmetric $X$ and $Y$ that place an arbitrary $n\times n$ block into an arbitrary $(i,j)$ block location in $Z$, then sums of such examples will achieve the desired result for an arbitrary $Z$.

For indices $i, j$ such that $1 \leq i \leq j \leq k$, let $\ell = j - i + 1$ and embed $\mathcal{H}^{(0)}_{\ell}(M)$ and $-\mathcal{H}^{(1)}_{\ell}(M)$ as principal submatrices in block rows and block columns $i$ through $j$ of block $k \times k$ zero matrices to get

$$(2.4)\qquad \widehat{X}_{ij} \boxplus\!\!\!\rightarrow \widehat{Y}_{ij} := \begin{array}{cc} & \begin{array}{cc} i & j \end{array} \\ \begin{array}{c} i \\ j \end{array} & \left[\begin{array}{c} \\ \boxed{\mathcal{H}^{(0)}_{\ell}(M)} \\ \\ \end{array}\right] \end{array} \boxplus\!\!\!\rightarrow \begin{array}{cc} & \begin{array}{cc} i & j \end{array} \\ \begin{array}{c} i \\ j \end{array} & \left[\begin{array}{c} \\ \boxed{-\mathcal{H}^{(1)}_{\ell}(M)} \\ \\ \end{array}\right] \end{array}$$

$$= \begin{array}{cc} & \begin{array}{cc} i & j+1 \end{array} \\ \begin{array}{c} i \\ j \end{array} & \left[\begin{array}{c} \\ \boxed{\mathcal{H}^{(0)}_{\ell}(M) \boxplus\!\!\!\rightarrow (-\mathcal{H}^{(1)}_{\ell}(M))} \\ \\ \end{array}\right] \end{array}$$

$$= E_{ji} \otimes M \quad (i \leq j).$$

Note that embedding $\mathcal{H}^{(0)}_{\ell}(M)$ and $-\mathcal{H}^{(1)}_{\ell}(M)$ as [illegible] block submatrices guarantees that $\widehat{X}_{ij}$ and $\widehat{Y}_{ij}$ are block symmetric. Similarly, defining the block symmetric matrices

$$(2.5)\qquad \widetilde{X}_{ij} = \begin{array}{cc} & \begin{array}{cc} i & j \end{array} \\ \begin{array}{c} i \\ j \end{array} & \left[\begin{array}{c} \\ \boxed{-\mathcal{H}^{(-1)}_{\ell}(P)} \\ \\ \end{array}\right] \end{array}, \quad \widetilde{Y}_{ij} = \begin{array}{cc} & \begin{array}{cc} i & j \end{array} \\ \begin{array}{c} i \\ j \end{array} & \left[\begin{array}{c} \\ \boxed{\mathcal{H}^{(0)}_{\ell}(P)} \\ \\ \end{array}\right] \end{array},$$

we have

$$(2.6)\qquad \widetilde{X}_{ij} \boxplus\!\!\!\rightarrow \widetilde{Y}_{ij} = E_{i,j+1} \otimes P \qquad (i \leq j).$$

Thus sums of these principally embedded versions of (2.2) and (2.3) can produce an arbitrary block $k \times (k+1)$ matrix $Z$ as the column-shifted sum of block symmetric $X$ and $Y$.

## 3. Block symmetric pencils and $\mathbb{DL}(P)$ for general $P$.

We now study the subspace of block symmetric pencils in $\mathbb{L}_1(P)$, which turns out to be the same as the space $\mathbb{DL}(P)$. This way of characterizing $\mathbb{DL}(P)$ leads to short proofs of some of its properties, as well as the identification of a useful basis.

### 3.1. Block symmetric pencils in $\mathbb{L}_1(P)$.

For a general polynomial $P$ we can use the results of section 2 to analyze the subspace

$$(3.1)\qquad \mathbb{B}(P) := \left\{\lambda X + Y \in \mathbb{L}_1(P) : X^{\mathcal{B}} = X,\ Y^{\mathcal{B}} = Y\right\}$$

of all block symmetric pencils in $\mathbb{L}_1(P)$. We will see in section 7 that almost all of these pencils are indeed linearizations for $P$.

Theorem 3.1. [illegible] $P(\lambda)$ [illegible] $k$ $\dim \mathbb{B}(P) = k$ [illegible] $v \in \mathbb{F}^k$ [illegible] $\mathbb{B}(P)$

[illegible]. Recalling that $\mathbb{L}_1$ is defined by (1.3), the theorem is proved if we can show that the map

$$
\begin{aligned}
\text{(3.2)}\qquad \mathbb{B}(P) &\xrightarrow{\mathcal{M}} \mathcal{V}_P := \{v \otimes P(\lambda) : v \in \mathbb{F}^k\},\\
L(\lambda) &\longmapsto L(\lambda)\,(\Lambda \otimes I_n)
\end{aligned}
$$

is a linear isomorphism.

First, recall from Lemma 2.4 that for any pencil $\lambda X + Y \in \mathbb{L}_1(P)$,

$$
(\lambda X + Y)(\Lambda \otimes I_n) = v \otimes P(\lambda) \iff X \boxplus\!\!\!\rightarrow Y = v \otimes [A_k\ A_{k-1}, \dots, A_0]. \tag{3.3}
$$

Thus $\lambda X + Y$ is in $\ker \mathcal{M}$ if and only if $X \boxplus\!\!\!\rightarrow Y = 0$. But $X$ and $Y$ are block symmetric, so by Lemma 2.5 we see that $\ker \mathcal{M} = \{0\}$, and hence $\mathcal{M}$ is 1-1.

To see that $\mathcal{M}$ is onto, let $v \otimes P(\lambda)$ with $v \in \mathbb{F}^k$ be an arbitrary element of $\mathcal{V}_P$. With $Z = v \otimes [A_k\ A_{k-1}, \dots, A_0]$, Lemma 2.5 shows that there exist block symmetric $X$ and $Y$ such that $X \boxplus\!\!\!\rightarrow Y = v \otimes [A_k\ A_{k-1}, \dots, A_0]$. Then by (3.3) we have $\mathcal{M}(\lambda X + Y) = v \otimes P(\lambda)$, showing that $\mathcal{M}$ is onto. □

### 3.2. Double ansatz pencils.

Our goal is now to show that $\mathbb{DL}(P) := \mathbb{L}_1(P) \cap \mathbb{L}_2(P) = \mathbb{B}(P)$. The inclusion $\mathbb{DL}(P) \subseteq \mathbb{B}(P)$, which says that all pencils $\lambda X + Y$ in $\mathbb{DL}(P)$ are block symmetric, can be deduced immediately from the following formulae for the blocks of $X$ and $Y$ in terms of the right ansatz vector $v$ [17, Thm. 5.3]:

$$
X_{ij} = v_{\max(i,j)} A_{k+1-\min(i,j)} + \sum_{\mu=1}^{\min(i-1,j-1)} \left(v_{j+i-\mu} A_{k+1-\mu} - v_\mu A_{k+1-j-i+\mu}\right),
$$

$$
Y_{ij} = \sum_{\mu=1}^{\min(i,j)} \left(v_\mu A_{k-j-i+\mu} - v_{j+i+1-\mu} A_{k+1-\mu}\right), \qquad i,j = 1\colon k.
$$

However, the derivation of these formulas is long and tedious. We present a shorter proof, based on first principles, of the stronger result $\mathbb{DL}(P) = \mathbb{B}(P)$.

LEMMA 3.2. [illegible] $P(\lambda)$ $\mathbb{B}(P) \subseteq \mathbb{DL}(P)$ [illegible] $L(\lambda) \in \mathbb{B}(P)$ [illegible]

[illegible]. For $L(\lambda) \in \mathbb{B}(P) \subset \mathbb{L}_1(P)$ with right ansatz vector $v$, we know from Theorem 2.2 that $L(\lambda)^{\mathcal{B}} = L(\lambda)$ is in $\mathbb{L}_2(P)$ with left ansatz vector $w = v$, and so $L(\lambda) \in \mathbb{DL}(P)$. □

Now we consider the special case of $\mathbb{DL}(P)$-pencils with $v = 0$, showing that in this case the left ansatz vector $w$ is forced to be 0 and the pencil is unique. Note that the definition of $\mathbb{DL}(P)$ does not require that $X$ and $Y$ are block symmetric, so we cannot appeal to Lemma 2.5 here.

LEMMA 3.3. [illegible] $L(\lambda) = \lambda X + Y \in \mathbb{DL}(P)$ [illegible] $v$ [illegible] $w$ [illegible] $v = 0$ [illegible] $w$ [illegible] 0 [illegible] $X = Y = 0$

[illegible]. We first show that the $\ell$th block column of $X$ and the $\ell$th coordinate of $w$ is zero for $\ell = 1\colon k$, by an induction on $\ell$.

Suppose that $\ell = 1$. From Lemma 2.4 we know that $X \boxplus\!\!\!\rightarrow Y = v \otimes [A_k\ A_{k-1}, \dots, A_0]$. Since $v = 0$ we have $X \boxplus\!\!\!\rightarrow Y = 0$, and hence the first block column of $X$ is zero. Now from Theorem 2.2, $L(\lambda)$ being in $\mathbb{L}_2(P)$ with left ansatz vector $w$ implies that $L(\lambda)^{\mathcal{B}} \in \mathbb{L}_1(P)$ with right ansatz vector $w$, which can be written in terms of the shifted sum as

$$X^{\mathcal{B}} \boxplus\!\!\rightarrow Y^{\mathcal{B}} = w \otimes [A_k\ A_{k-1}, \dots, A_0]. \tag{3.4}$$

The $(1,1)$-block of the right-hand side of (3.4) is $w_1 A_k$, while on the left-hand side the $(1,1)$-block of $X^{\mathcal{B}} \boxplus\!\!\rightarrow Y^{\mathcal{B}}$ is the same as the $(1,1)$-block of $X$. Hence $w_1 A_k = 0$. But the leading coefficient $A_k$ of $P(\lambda)$ is nonzero by assumption, so $w_1 = 0$.

Now suppose that the $\ell$th block column of $X$ is zero and that $w_\ell = 0$. Then by (3.4) the $\ell$th block row of $X^{\mathcal{B}} \boxplus\!\!\rightarrow Y^{\mathcal{B}}$ is zero. Since the $\ell$th block row of $X^{\mathcal{B}}$ is zero, the $\ell$th block row of $Y^{\mathcal{B}}$, or, equivalently, the $\ell$th block column of $Y$, must also be zero. Combining this with $X \boxplus\!\!\rightarrow Y = 0$ implies that the $(\ell+1)$st block column of $X$ is zero. Now equating the $(\ell+1, 1)$-blocks of both sides of (3.4) gives $w_{\ell+1} A_k = 0$, and hence $w_{\ell+1} = 0$. This concludes the induction, and shows that $X = 0$ and $w = 0$.

Finally, $X = 0$ and $X \boxplus\!\!\rightarrow Y = 0$ implies $Y = 0$, completing the proof. □

We can now characterize $\mathbb{DL}(P)$ and give a precise description of all right/left ansatz vector pairs $(v, w)$ that can be realized by some $\mathbb{DL}(P)$-pencil. The latter has already been done in [17, Thm. 5.3], but with a much longer derivation.

THEOREM 3.4. [illegible] $P(\lambda)$ [illegible] $k$ $\mathbb{DL}(P) = \mathbb{B}(P)$ [illegible] $L(\lambda) \in \mathbb{DL}(P)$ [illegible] $v$ [illegible] $w$ [illegible] $v = w$ [illegible] $L$ [illegible] $\dim \mathbb{DL}(P) = k$ [illegible] $v \in \mathbb{F}^k$ [illegible] $\mathbb{DL}(P)$

[illegible] Consider the linear map $\mathbb{DL}(P) \longrightarrow \mathbb{F}^k$ that associates to any pencil $L(\lambda) \in \mathbb{DL}(P)$ its right ansatz vector $v \in \mathbb{F}^k$. By Lemma 3.3 this map is injective, so that $\dim \mathbb{DL}(P) \le k$. But $\mathbb{B}(P) \subseteq \mathbb{DL}(P)$ by Lemma 3.2, and $\dim \mathbb{B}(P) = k$ by Theorem 3.1, so $\mathbb{B}(P) = \mathbb{DL}(P)$. The rest of the theorem follows from the properties of $\mathbb{B}(P)$ in Theorem 3.1 and Lemma 3.2. □

The equality $\mathbb{DL}(P) = \mathbb{B}(P)$ can be thought of as saying that the pencils in $\mathbb{DL}(P)$ are [illegible] structured: they have block symmetry as well as the eigenvector recovery properties that were the original motivation for their definition.

**3.3. The standard basis for $\mathbb{B}(P)$.** The isomorphism established in the proof of Theorem 3.1 immediately suggests the possibility that the basis for $\mathbb{B}(P)$ corresponding (via the map $\mathcal{M}$ in (3.2)) to the standard basis $\{e_1, \dots, e_k\}$ for $\mathbb{F}^k$ may be especially simple and useful. In this section we derive a general formula for these "standard basis pencils" in $\mathbb{B}(P)$ as a corollary of the shifted sum construction used in section 2 to build block symmetric solutions of the equation $X \boxplus\!\!\rightarrow Y = Z$. These pencils are of course also a basis for $\mathbb{DL}(P)$, since $\mathbb{DL}(P) = \mathbb{B}(P)$.

In light of Lemma 2.4, then, our goal is to construct for each $1 \le m \le k$ a block symmetric pencil $\lambda X_m + Y_m$ such that

$$X_m \boxplus\!\!\rightarrow Y_m \;=\; e_m \otimes [\, A_k\ A_{k-1}, \dots, A_0 \,]. \tag{3.5}$$

This is most easily done in two steps. First we show how to achieve the first $m$ block columns in the desired shifted sum, i.e., how to get $e_m \otimes [\, A_k, \dots, A_{k-m+1}\ \ 0, \dots, 0 \,]$. Then the last $k-m+1$ block columns $e_m \otimes [\, 0, \dots, 0\ A_{k-m}, \dots, A_1\ A_0 \,]$ are produced by a related but slightly different construction. We use the following notation for principal block submatrices, adapted from [12]: for a block $k \times k$ matrix $X$ and index set $\alpha \subseteq \{1, 2, \dots, k\}$, $X(\alpha)$ will denote the principal block submatrix lying in the block rows and block columns with indices in $\alpha$.

To get the first $m$ block columns in the desired shifted sum we repeatedly use the construction in (2.4) to build block $k \times k$ matrices $\widehat{X}_m$ and $\widehat{Y}_m$, embedding once in each of the principal block submatrices $\widehat{X}_m(\alpha_i)$ and $\widehat{Y}_m(\alpha_i)$ for the index sets $\alpha_i = \{i, i+1, \dots, m\}$, $i = 1\colon m$. Accumulating these embedded submatrices, we obtain

$$
\widehat{X}_m = \begin{array}{cl}
\begin{array}{ccccc} & & & m & \phantom{0} \end{array} & \\
\left[\begin{array}{cccc|c}
 & & & A_k & \\
 & & \iddots & A_{k-1} & \\
 & \iddots & \iddots & \vdots & 0 \\
\iddots & \iddots & & \vdots & \\
A_k & A_{k-1} & \cdots\cdots & A_{k-m+1} & \\ \hline
 & & & & \\
 & 0 & & & 0 \\
 & & & &
\end{array}\right] & \begin{array}{l} \\ \\ \\ \\ m, \\ \\ \\ \\ \end{array}
\end{array}
$$

$$
\widehat{Y}_m = -\begin{array}{cl}
\begin{array}{ccccc} & & & & m \end{array} & \\
\left[\begin{array}{ccccc|c}
 & & & A_k & 0 & \\
 & & \iddots & A_{k-1} & 0 & \\
 & \iddots & \iddots & \vdots & \vdots & 0 \\
A_k & A_{k-1} & \cdots & A_{k-m+2} & 0 & \\
0 & \cdots & \cdots & 0 & 0 & \\ \hline
 & & & & & \\
 & & 0 & & & 0 \\
 & & & & &
\end{array}\right] & \begin{array}{l} \\ \\ \\ \\ \\ m, \\ \\ \\ \\ \end{array}
\end{array}
$$

with the property that $\widehat{X}_m \boxplus\!\!\!\rightarrow \widehat{Y}_m = e_m \otimes [\, A_k, \dots, A_{k-m+1} \;\; 0, \dots, 0 \,]$.

To obtain the last $k-m+1$ columns we use the construction outlined in (2.5) and (2.6) $k-m+1$ times to build block $k \times k$ matrices $\widetilde{X}_m$ and $\widetilde{Y}_m$ , embedding once in each of the principal block submatrices $\widetilde{X}_m(\beta_j)$ and $\widetilde{Y}_m(\beta_j)$ for the index sets $\beta_j = \{m, m+1, \dots, j\}$, $j = m\colon k$, which yields

$$
\widetilde{X}_m = -\begin{array}{cl}
\begin{array}{cccccc} \phantom{0} & m & & & & k \end{array} & \\
\left[\begin{array}{c|ccccc}
0 & & & 0 & & \\ \hline
 & 0 & 0 & \cdots\cdots & & 0 \\
 & 0 & A_{k-m-1} & \iddots & A_1 & A_0 \\
0 & \vdots & \vdots & \iddots & \iddots & \\
 & \vdots & A_1 & \iddots & & \\
 & 0 & A_0 & & &
\end{array}\right] & \begin{array}{l} \\ \\ m, \\ \\ \\ \\ k \end{array}
\end{array}
$$

$$
\widetilde{Y}_m = \begin{array}{cl}
\begin{array}{ccccc} \phantom{0} & m & & & k \end{array} & \\
\left[\begin{array}{c|cccc}
0 & & & 0 & \\ \hline
 & A_{k-m} & \cdots\cdots & A_1 & A_0 \\
 & \vdots & & \iddots & A_0 \\
0 & \vdots & \iddots & \iddots & \\
 & A_1 & \iddots & & \\
 & A_0 & & &
\end{array}\right] & \begin{array}{l} \\ \\ m, \\ \\ \\ \\ k \end{array}
\end{array}
$$

satisfying $\widetilde{X}_m \boxplus\!\!\rightarrow \widetilde{Y}_m = e_m \otimes [\,0, \dots, 0\ A_{k-m}, \dots, A_1\ A_0\,]$. With $X_m := \widehat{X}_m + \widetilde{X}_m$ and $Y_m := \widehat{Y}_m + \widetilde{Y}_m$ we have $X_m \boxplus\!\!\rightarrow Y_m = e_m \otimes [\,A_k\ A_{k-1}, \dots, A_1\ A_0\,]$, so $\lambda X_m + Y_m$ is the $m$th standard basis pencil for $\mathbb{B}(P)$.

A more concise way to express the $m$th standard basis pencil uses the following block Hankel matrices. Let $\mathcal{L}_j(P(\lambda))$ denote the lower block antitriangular, block Hankel, block $j \times j$ matrix

$$\mathcal{L}_j(P(\lambda)) := \begin{bmatrix} & & & A_k \\ & & \cdot^{\cdot^{\cdot}} & A_{k-1} \\ & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ A_k & A_{k-1} & \dots & A_{k-j+1} \end{bmatrix} \tag{3.6}$$

formed from the first $j$ matrix coefficients $A_k, A_{k-1}, \dots, A_{k-j+1}$ of $P(\lambda)$. Similarly, let $\mathcal{U}_j(P(\lambda))$ denote the upper block antitriangular, block Hankel, block $j \times j$ matrix

$$\mathcal{U}_j(P(\lambda)) := \begin{bmatrix} A_{j-1} & \dots & A_1 & A_0 \\ \vdots & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \\ A_1 & \cdot^{\cdot^{\cdot}} & & \\ A_0 & & & \end{bmatrix} \tag{3.7}$$

formed from the last $j$ matrix coefficients $A_{j-1}, A_{j-2}, \dots, A_1, A_0$ of $P(\lambda)$. Then the block symmetric matrices $X_m$ and $Y_m$ in the $m$th standard basis pencil ($m = 1\colon k$) can be neatly expressed as direct sums of block Hankel matrices:

$$X_m = X_m(P(\lambda)) = \begin{bmatrix} \mathcal{L}_m(P(\lambda)) & 0 \\ 0 & -\mathcal{U}_{k-m}(P(\lambda)) \end{bmatrix}, \tag{3.8a}$$

$$Y_m = Y_m(P(\lambda)) = \begin{bmatrix} -\mathcal{L}_{m-1}(P(\lambda)) & 0 \\ 0 & \mathcal{U}_{k-m+1}(P(\lambda)) \end{bmatrix}. \tag{3.8b}$$

($\mathcal{L}_j$ and $\mathcal{U}_j$ are taken to be void when $j = 0$.) From (3.8) it now becomes obvious that the coefficient matrices in successive standard basis pencils are closely related:

$$Y_m(P(\lambda)) \;=\; -X_{m-1}(P(\lambda)), \qquad m = 1\colon k. \tag{3.9}$$

Thus we have the following explicit formula for the standard basis pencils in $\mathbb{B}(P)$.

THEOREM 3.5. [illegible] $P(\lambda)$ [illegible] $k$ [illegible] $m = 1\colon k$ [illegible] $\mathbb{B}(P)$ [illegible] $e_m$ [illegible] $\lambda X_m - X_{m-1}$ [illegible] $X_m$ [illegible] (3.8a)

The standard basis pencils in $\mathbb{B}(P)$ for general polynomials of degree 2 and 3 are listed in Tables 3.1 and 3.2, where the partitioning from (3.8) is shown in each

TABLE 3.1

*Block symmetric standard basis for the quadratic* $P(\lambda) = \lambda^2 A + \lambda B + C$.

| $v$ | $L(\lambda) \in \mathbb{B}(P)$ |
|---|---|
| $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ | $\lambda \left[\begin{array}{c\|c} A & 0 \\ \hline 0 & -C \end{array}\right] + \begin{bmatrix} B & C \\ C & 0 \end{bmatrix}$ |
| $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ | $\lambda \begin{bmatrix} 0 & A \\ A & B \end{bmatrix} + \left[\begin{array}{c\|c} -A & 0 \\ \hline 0 & C \end{array}\right]$ |

Table 3.2
*Block symmetric standard basis for the cubic* $P(\lambda) = \lambda^3 A + \lambda^2 B + \lambda C + D$.

<table>
<tr><th>$v$</th><th>$L(\lambda) \in \mathbb{B}(P)$</th></tr>
<tr><td>$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$</td><td>$\lambda \left[\begin{array}{c|cc} A & 0 & 0 \\ \hline 0 & -C & -D \\ 0 & -D & 0 \end{array}\right] + \begin{bmatrix} B & C & D \\ C & D & 0 \\ D & 0 & 0 \end{bmatrix}$</td></tr>
<tr><td>$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$</td><td>$\lambda \left[\begin{array}{cc|c} 0 & A & 0 \\ A & B & 0 \\ \hline 0 & 0 & -D \end{array}\right] + \left[\begin{array}{c|cc} -A & 0 & 0 \\ \hline 0 & C & D \\ 0 & D & 0 \end{array}\right]$</td></tr>
<tr><td>$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$</td><td>$\lambda \begin{bmatrix} 0 & 0 & A \\ 0 & A & B \\ A & B & C \end{bmatrix} + \left[\begin{array}{cc|c} 0 & -A & 0 \\ -A & -B & 0 \\ \hline 0 & 0 & D \end{array}\right]$</td></tr>
</table>

case. As an immediate consequence we have, for the important case of quadratics $P(\lambda) = \lambda^2 A + \lambda B + C$, the following description of all block symmetric pencils in $\mathbb{L}_1(P)$:

$$\mathbb{B}(P) = \left\{ L(\lambda) = \lambda \begin{bmatrix} v_1 A & v_2 A \\ v_2 A & v_2 B - v_1 C \end{bmatrix} + \begin{bmatrix} v_1 B - v_2 A & v_1 C \\ v_1 C & v_2 C \end{bmatrix} \; : \; v \in \mathbb{C}^2 \right\}.$$

## 4. Other constructions of block symmetric linearizations.

Several other methods for constructing block symmetric linearizations of matrix polynomials have appeared previously in the literature.

Antoniou and Vologiannidis [1] have recently found new companion-like linearizations for general matrix polynomials $P$ by generalizing Fiedler's results [7] on a factorization of the companion matrix of a scalar polynomial and certain of its permutations. From this finite family of $\frac{1}{6}(2 + \deg P)!$ pencils, all of which are linearizations, they identify one distinguished pencil that is Hermitian whenever $P$ is Hermitian. But this example has structure even for general $P$: it is block symmetric. Indeed, it provides a simple example of a block symmetric linearization for $P(\lambda)$ that is not in $\mathbb{B}(P)$. In the case of a cubic polynomial $P(\lambda) = \lambda^3 A + \lambda^2 B + \lambda C + D$, the pencil is

$$L(\lambda) = \lambda \begin{bmatrix} A & 0 & 0 \\ 0 & 0 & I \\ 0 & I & C \end{bmatrix} + \begin{bmatrix} B & -I & 0 \\ -I & 0 & 0 \\ 0 & 0 & D \end{bmatrix}. \tag{4.1}$$

Using the column-shifted sum it is easy to see that $L(\lambda)$ is not in $\mathbb{L}_1(P)$, and hence not in $\mathbb{B}(P)$.

Contrasting with the "permuted factors" approach of [1], [7] and the additive construction used in this paper, is a third "multiplicative" method for generating block symmetric linearizations described by Lancaster in [14], [15]. In [14] only scalar polynomials $p(\lambda) = a_k\lambda^k + \cdots + a_1\lambda + a_0$ are considered; the starting point is the companion matrix of $p(\lambda)$,

$$C = \begin{bmatrix} -a_k^{-1} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \begin{bmatrix} a_{k-1} & a_{k-2} & \dots & a_0 \\ 1 & 0 & \dots & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix}, \tag{4.2}$$

and the associated pencil $\lambda I - C$. Lancaster's idea is to seek a nonsingular symmetric matrix $B$ such that $BC$ is symmetric, thus providing a symmetric linearization $B(\lambda I - C) = \lambda B - BC$ for $p(\lambda)$. That such a $B$ can always be found follows from a standard result in matrix theory [12, Cor. 4.4.11]. Lancaster shows further that $B$ and $BC$ symmetric imply $BC^j$ is symmetric for all $j \geq 1$; thus $BC^{j-1}(\lambda I - C) = \lambda BC^{j-1} - BC^j$ is a symmetric pencil for any $j \geq 1$, and for $j \geq 2$ it is a linearization of $p(\lambda)$ if $a_0 \neq 0$. This strategy is realized in [14] with the particular choice of symmetric (Hankel) matrix

$$B = \begin{bmatrix} & & & a_k \\ & & \iddots & a_{k-1} \\ & \iddots & \iddots & \vdots \\ a_k & a_{k-1} & \dots & a_1 \end{bmatrix}, \tag{4.3}$$

which is nonsingular since $a_k \neq 0$, and it is observed that this particular $B$ gives symmetric pencils $\lambda BC^{j-1} - BC^j$ with an especially simple form for $1 \leq j \leq k$, though apparently with a much more complicated form for $j > k$.

It is easy to see that these symmetric pencils, constructed for scalar polynomials $p(\lambda)$, can be immediately extended to block symmetric pencils for general matrix polynomials $P(\lambda)$ simply by formally replacing the scalar coefficients of $p(\lambda)$ in $B, BC, BC^2, \dots$ by the matrix coefficients of $P(\lambda)$. This has been done in [15, sect. 4.2] and [8]. Garvey et al. [8] go even further with these block symmetric pencils, using them as a foundation for defining a new class of isospectral transformations on matrix polynomials.

Since Lancaster's construction of pencils is so different from ours there is no a priori reason to expect any connection between his pencils and the pencils in $\mathbb{DL}(P)$. The next result shows, rather surprisingly, that the first $k$ pencils in Lancaster's sequence generate $\mathbb{DL}(P)$.

THEOREM 4.1. [illegible] $P(\lambda)$ [illegible] $k$ [illegible] $\lambda BC^{k-m} - BC^{k-m+1}$ [illegible] $B$ [illegible] $C$ [illegible] (4.2) [illegible] (4.3) [illegible] $\lambda X_m - X_{m-1}$ [illegible] $m$ [illegible] $\mathbb{DL}(P)$ [illegible] $m = 1\colon k$

[illegible]. We have to show that $X_m = BC^{k-m}$, $m = 0\colon k$, where $X_m$ is given by (3.8a). For notational simplicity we will carry out the proof for a scalar polynomial; the same proof applies to a matrix polynomial with only minor changes in notation. The $m = k$ case, $X_k(p(\lambda)) = \mathcal{L}_k(p(\lambda)) = B$, is immediate from (3.6), (3.8), and (4.3). The rest follow inductively (downward) from the relation $X_{m-1}(p(\lambda)) = X_m(p(\lambda)) \cdot C$, which we now proceed to show holds for $m = 1\colon k$.

To see that $X_m C = X_{m-1}$, or equivalently that

$$\begin{bmatrix} \mathcal{L}_m(p(\lambda)) & 0 \\ 0 & -\mathcal{U}_{k-m}(p(\lambda)) \end{bmatrix} C = \begin{bmatrix} \mathcal{L}_{m-1}(p(\lambda)) & 0 \\ 0 & -\mathcal{U}_{k-m+1}(p(\lambda)) \end{bmatrix}$$

holds for $1 \leq m \leq k$, it will be convenient to rewrite the companion matrix (4.2) in the form

$$C = N_k^T - a_k^{-1} \begin{bmatrix} a_{k-1} & a_{k-2} & \dots & a_0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} = N_k^T - a_k^{-1} e_1 \begin{bmatrix} a_{k-1} & a_{k-2} & \dots & a_0 \end{bmatrix},$$

where $N_k$ is defined in (2.1). Then

$$\begin{aligned} X_m(p(\lambda))C &= X_m(p(\lambda))N_k^T - a_k^{-1}X_m(p(\lambda))\, e_1 \begin{bmatrix} a_{k-1} & a_{k-2} & \dots & a_0 \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{L}_m(p(\lambda)) & 0 \\ 0 & -\mathcal{U}_{k-m}(p(\lambda)) \end{bmatrix} N_k^T - e_m \begin{bmatrix} a_{k-1} & a_{k-2} & \dots & a_0 \end{bmatrix}. \end{aligned}$$

In the first term, postmultiplication by $N_k^T$ has the effect of shifting the columns to the left by one (and losing the first column), thus giving

$$\begin{aligned} X_m(p(\lambda))C &= \begin{bmatrix} \mathcal{L}_{m-1}(p(\lambda)) & 0 & 0 \\ a_{k-1} \dots a_{k-m+1} & 0 & 0 \\ 0 & -\mathcal{U}_{k-m}(p(\lambda)) & 0 \end{bmatrix} \\ &\quad - \begin{bmatrix} 0 & 0 & 0 \\ a_{k-1} \dots a_{k-m+1} & a_{k-m} \dots a_1 & a_0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \left[\begin{array}{c|cc} \mathcal{L}_{m-1}(p(\lambda)) & 0 & 0 \\ \hline 0 & -a_{k-m} \dots -a_1 & -a_0 \\ 0 & -\mathcal{U}_{k-m}(p(\lambda)) & 0 \end{array}\right] \\ &= \begin{bmatrix} \mathcal{L}_{m-1}(p(\lambda)) & 0 \\ 0 & -\mathcal{U}_{k-m+1}(p(\lambda)) \end{bmatrix} = X_{m-1}(p(\lambda)). \end{aligned}$$

This completes the inductive step of the proof. □

**5. Symmetric pencils in $\mathbb{L}_1(P)$ for symmetric $P$.** We now return to the problem that originally motivated the investigation in this paper, that of systematically finding large sets of symmetric linearizations for symmetric polynomials, $P(\lambda) = P(\lambda)^T$. Our strategy is first to characterize the set

$$\mathbb{S}(P) := \left\{ \lambda X + Y \in \mathbb{L}_1(P) : X^T = X,\, Y^T = Y \right\} \tag{5.1}$$

of all symmetric pencils in $\mathbb{L}_1(P)$ when $P$ is symmetric, and then later in section 7 to show that almost all of these symmetric pencils are indeed linearizations for $P$.

We begin with a result for symmetric polynomials reminiscent of Theorem 2.2, but using transpose rather than block transpose.

Lemma 5.1. [illegible] $P(\lambda)$ [illegible] $L(\lambda) \in \mathbb{L}_1(P)$ [illegible] $v$ [illegible] $L^T(\lambda) \in \mathbb{L}_2(P)$ [illegible] $w = v$ [illegible] $L(\lambda) \in \mathbb{L}_2(P)$ [illegible] $v$ [illegible] $L^T(\lambda) \in \mathbb{L}_1(P)$ [illegible] $v$

[illegible]. Suppose $L(\lambda) \in \mathbb{L}_1(P)$ with right ansatz vector $v$. Then

$$\big(L(\lambda)(\varLambda \otimes I)\big)^T = \big(v \otimes P(\lambda)\big)^T \Longrightarrow (\varLambda^T \otimes I)L^T(\lambda) = v^T \otimes P(\lambda).$$

Thus $L^T(\lambda) \in \mathbb{L}_2(P)$ with left ansatz vector $v$. The proof of the second statement is analogous. □

We characterize the space $\mathbb{S}(P)$ in the next result by relating it to the previously developed space $\mathbb{DL}(P)$, which we already know equals $\mathbb{B}(P)$.

Theorem 5.2. [illegible] $P(\lambda)$ $\mathbb{S}(P) = \mathbb{DL}(P)$

[illegible] Suppose $L(\lambda) \in \mathbb{S}(P) \subseteq \mathbb{L}_1(P)$ with right ansatz vector $v$. Then by Lemma 5.1 we know that $L^T(\lambda) = L(\lambda)$ is in $\mathbb{L}_2(P)$ with left ansatz vector $v$, and so $L(\lambda) \in \mathbb{DL}(P)$. Thus $\mathbb{S}(P) \subseteq \mathbb{DL}(P)$.

By Lemma 5.1, $L(\lambda) \in \mathbb{DL}(P)$ with right/left ansatz vector $v$ implies that $L^T(\lambda) \in \mathbb{DL}(P)$ with left/right ansatz vector $v$. But by Theorem 3.4 pencils in $\mathbb{DL}(P)$ are uniquely determined by their ansatz vector, so $L(\lambda) \equiv L^T(\lambda)$, and hence $\mathbb{DL}(P) \subseteq \mathbb{S}(P)$. Therefore $\mathbb{DL}(P) = \mathbb{S}(P)$. ☐

Once again one may refer to Tables 3.1 and 3.2 for examples of what are in effect [illegible] structured pencils whenever $P$ is symmetric. Recall, however, that there are symmetric linearizations for $P$ that are not in $\mathbb{S}(P)$: $L$ in (4.1) is not in $\mathbb{S}(P)$, but is a symmetric linearization for any symmetric cubic $P$.

## 6. Hermitian pencils in $\mathbb{L}_1(P)$ for Hermitian $P$.

For a Hermitian matrix polynomial $P(\lambda)$ of degree $k$, that is, $P(\lambda)^* = P(\overline{\lambda})$, let

$$\mathbb{H}(P) := \left\{ \lambda X + Y \in \mathbb{L}_1(P) : X^* = X,\ Y^* = Y \right\} \tag{6.1}$$

denote the set of all Hermitian pencils in $\mathbb{L}_1(P)$. A priori the right ansatz vector $v$ of a pencil in $\mathbb{H}(P)$ might be any vector in $\mathbb{C}^k$, since $P$ is a complex polynomial. However, we will see that any such $v$ must in fact be real, and furthermore that any Hermitian pencil in $\mathbb{L}_1(P)$ must be in $\mathbb{DL}(P)$.

THEOREM 6.1. [illegible] $P(\lambda)$ [illegible] $\mathbb{H}(P)$ [illegible] $\mathbb{DL}(P)$ [illegible] $v \in \mathbb{R}^k$ [illegible] $\mathbb{H}(P)$

[illegible] Suppose $L(\lambda) \in \mathbb{H}(P) \subset \mathbb{L}_1(P)$ with right ansatz vector $v$, so that $L(\lambda)(\Lambda \otimes I) = v \otimes P(\lambda)$. Then, since $P$ and $L$ are Hermitian,

$$\big(L(\lambda)(\Lambda \otimes I)\big)^* = \big(v \otimes P(\lambda)\big)^* \Longrightarrow (\overline{\Lambda}^T \otimes I)L(\overline{\lambda}) \;=\; \overline{v}^T \otimes P(\overline{\lambda}).$$

This last equation holds for all $\lambda$, so we may replace $\overline{\lambda}$ by $\lambda$ to get $(\Lambda^T \otimes I)L(\lambda) = \overline{v}^T \otimes P(\lambda)$, so that $L(\lambda) \in \mathbb{L}_2(P)$ with left ansatz vector $w = \overline{v}$. Thus $L(\lambda) \in \mathbb{DL}(P)$ and $\mathbb{H}(P) \subseteq \mathbb{DL}(P)$. But by Theorem 3.4 the right and left ansatz vectors of any $\mathbb{DL}(P)$-pencil must be equal. So $v = \overline{v}$, which means $v \in \mathbb{R}^k$.

Conversely, for an arbitrary $v \in \mathbb{R}^k$ let $H(\lambda)$ be the unique pencil in $\mathbb{DL}(P)$ with right/left ansatz vector $v$. By arguments analogous to those used in Lemma 5.1 it is straightforward to show that for Hermitian $P$, $L(\lambda) \in \mathbb{DL}(P)$ with any right/left ansatz vector $v \in \mathbb{C}^k$ implies that $L^*(\lambda) := \lambda X^* + Y^* \in \mathbb{DL}(P)$ with left/right ansatz vector $\overline{v}$. But $H(\lambda)$ has a real ansatz vector $v$, so $H^*(\lambda) \in \mathbb{DL}(P)$ with exactly the same ansatz vector $v$. Thus the uniqueness of $\mathbb{DL}(P)$-pencils established in Theorem 3.4 implies that $H(\lambda) \equiv H^*(\lambda)$, i.e., $H(\lambda) \in \mathbb{H}(P)$. ☐

## 7. Almost all pencils in $\mathbb{B}(P)$, $\mathbb{DL}(P)$, $\mathbb{S}(P)$, and $\mathbb{H}(P)$ are linearizations.

The remaining fundamental issue is the question of which pencils in the subspaces $\mathbb{B}(P)$, $\mathbb{DL}(P)$, $\mathbb{S}(P)$, and $\mathbb{H}(P)$ are actually linearizations for $P$ when $P$ is regular. Some answers to this question are already known. First, a pencil $L$ in $\mathbb{L}_1(P)$ or $\mathbb{L}_2(P)$ is a linearization precisely when $L$ is a regular pencil [17, Thm. 4.3]. Second, for each of $\mathbb{L}_1(P)$, $\mathbb{L}_2(P)$, and $\mathbb{DL}(P)$ it is known that almost all pencils are linearizations, where "almost all" means all except for a closed, nowhere dense set of measure zero [17, Thms. 4.7, 6.8]. Because of Theorems 3.4 and 5.2, the same conclusion follows immediately for $\mathbb{B}(P)$, and for $\mathbb{S}(P)$ when $P$ is symmetric. However, for $\mathbb{H}(P)$ the possible ansatz vectors lie in $\mathbb{R}^k$, a closed, nowhere dense set of measure zero in $\mathbb{C}^k$

(the ansatz vector set of $\mathbb{DL}(P)$ when $P$ is Hermitian), so we cannot immediately deduce an "almost all" result for $\mathbb{H}(P)$. Some further analysis is therefore needed.

To a vector $v = [v_1, v_2, \dots, v_k]^T \in \mathbb{F}^k$ associate the scalar polynomial

$$\mathsf{p}(x; v) = v_1 x^{k-1} + v_2 x^{k-2} + \cdots + v_{k-1} x + v_k,$$

referred to as the "$\mathsf{v}$-polynomial" of the vector $v$. We adopt the convention that $\mathsf{p}(x; v)$ has a root at $\infty$ whenever $v_1 = 0$. The following result provides a condition that $L \in \mathbb{DL}(P)$ be a linearization of $P$.

Theorem 7.1 (eigenvalue exclusion theorem [17, Thm. 6.7]). [illegible] $P(\lambda)$ [illegible] $L(\lambda) \in \mathbb{DL}(P)$ [illegible] $v$ [illegible] $L(\lambda)$ [illegible] $P(\lambda)$ [illegible] $\mathsf{v}$ [illegible] $\mathsf{p}(x; v)$ [illegible] $P(\lambda)$

With the aid of this result we can establish the desired genericity statement.

Theorem 7.2 (linearizations are generic in $\mathbb{H}(P)$). [illegible] $P(\lambda)$ [illegible] $v \in \mathbb{R}^k$ [illegible] $\mathbb{H}(P)$ [illegible]

[illegible] Recall that the resultant $\operatorname{res}(f, g)$ of two polynomials $f(x)$ and $g(x)$ is a polynomial in the coefficients of $f$ and $g$ with the property that $\operatorname{res}(f, g) = 0$ if and only if $f(x)$ and $g(x)$ have a common (finite) root [22, p. 248], [26]. Now consider $r = \operatorname{res}\bigl(\mathsf{p}(x; v), \det P(x)\bigr)$, which, because $P$ is Hermitian and fixed, can be viewed as a real polynomial $r(v_1, v_2, \dots, v_k)$ in the components of $v \in \mathbb{R}^k$. The zero set $\mathcal{Z}(r) = \bigl\{ v \in \mathbb{R}^k : r(v_1, v_2, \dots, v_k) = 0 \bigr\}$, then, is exactly the set of all $v \in \mathbb{R}^k$ for which some finite root of $\mathsf{p}(x; v)$ is an eigenvalue of $P(\lambda)$. Recall that by our convention the $\mathsf{v}$-polynomial $\mathsf{p}(x; v)$ has $\infty$ as a root exactly for $v \in \mathbb{R}^k$ lying in the hyperplane $v_1 = 0$. Thus by Theorem 7.1 the set of vectors $v \in \mathbb{R}^k$ for which the corresponding pencil $L(\lambda) \in \mathbb{H}(P) \subset \mathbb{DL}(P)$ is [illegible] a linearization of $P(\lambda)$ is either the proper (real) algebraic set $\mathcal{Z}(r)$, or the union of two proper (real) algebraic sets, $\mathcal{Z}(r)$ and the hyperplane $v_1 = 0$. But the union of any finite number of proper (real) algebraic sets is always a closed, nowhere dense set of measure zero in $\mathbb{R}^k$. ☐

## 8. Conclusions.

We have revisited $\mathbb{DL}(P)$, the space of double ansatz pencils introduced in [17], proving that it is the same as the set of block symmetric pencils in the right ansatz space $\mathbb{L}_1(P)$. Our alternative characterization of $\mathbb{DL}(P)$ shows that even unstructured matrix polynomials admit linearizations that are symmetric at the block level, while simultaneously possessing the $\mathbb{DL}(P)$ property of revealing both left and right eigenvectors of $P$.

Our analysis shows how to find all the symmetric pencils in $\mathbb{L}_1(P)$ for a symmetric matrix polynomial $P$: these are precisely the pencils in $\mathbb{DL}(P)$. For Hermitian $P$, the Hermitian pencils in $\mathbb{L}_1(P)$ correspond to the double ansatz pencils that have a real ansatz vector. Almost all pencils in each of these vector spaces have been shown to be linearizations.

**Acknowledgment.** We thank the referees for perceptive comments that led to an improvement in the presentation.

## REFERENCES

[1] E. N. Antoniou and S. Vologiannidis, *A new family of companion forms of polynomial matrices*, Electron. J. Linear Algebra, 11 (2004), pp. 78–87.

[2] Z. Bai, J. W. Demmel, J. J. Dongarra, A. Ruhe, and H. A. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Software Environ. Tools 11, SIAM, Philadelphia, 2000.

[3] D. A. Bini, L. Gemignani, and F. Tisseur, *The Ehrlich–Aberth method for the nonsymmetric tridiagonal eigenvalue problem*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 153–175.

[4] M. A. Brebner and J. Grad, *Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices $A$ and $B$ computed by reduction to a pseudosymmetric form and the $HR$ process*, Linear Algebra Appl., 43 (1982), pp. 99–118.

[5] T. J. Bridges and P. J. Morris, *Differential eigenvalue problems in which the parameter appears nonlinearly*, J. Comput. Phys., 55 (1984), pp. 437–460.

[6] A. Bunse-Gerstner, *An analysis of the $HR$ algorithm for computing the eigenvalues of a matrix*, Linear Algebra Appl., 35 (1981), pp. 155–173.

[7] M. Fiedler, *A note on companion matrices*, Linear Algebra Appl., 372 (2003), pp. 325–331.

[8] S. Garvey, U. Prells, M. I. Friswell, and Z. Chen, *General isospectral flows for linear dynamic systems*, Linear Algebra Appl., 385 (2004), pp. 335–368.

[9] N. J. Higham, R.-C. Li, and F. Tisseur, *Backward error of polynomial eigenproblems solved by linearization*, MIMS EPrint 2006.137, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2006.

[10] N. J. Higham, D. S. Mackey, and F. Tisseur, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), 1005–1028.

[11] N. J. Higham, F. Tisseur, and P. M. Van Dooren, *Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems*, Linear Algebra Appl., 351–352 (2002), pp. 455–474.

[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[13] T.-M. Hwang, W.-W. Li, J.-L. Liu, and W. Wang, *Jacobi-Davidson methods for cubic eigenvalue problems*, Numer. Linear Algebra Appl., 12 (2005), pp. 605–624.

[14] P. Lancaster, *Symmetric transformations of the companion matrix*, NABLA: Bulletin of the Malayan Math. Soc., 8 (1961), pp. 146–148.

[15] P. Lancaster, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, UK, 1966. Reprinted by Dover, New York, 2002.

[16] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed., Academic Press, London, 1985.

[17] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), 971–1004.

[18] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), 1029–1051.

[19] V. Mehrmann and D. Watkins, *Polynomial eigenvalue problems with Hamiltonian structure*, Electron. Trans. Numer. Anal., 13 (2002), pp. 106–118.

[20] B. N. Parlett and H. C. Chen, *Use of indefinite pencils for computing damped natural modes*, Linear Algebra Appl., 140 (1990), pp. 53–88.

[21] H. Rutishauser, *Solution of eigenvalue problems with the LR-transformation*, in Further Contributions to the Solution of Simultaneous Linear Equations and the Determination of Eigenvalues, Nat. Bur. Standards Appl. Math. Ser. 49, United States Department of Commerce, Washington, D.C., 1958, pp. 47–81.

[22] H. Stetter, *Numerical Polynomial Algebra*, SIAM, Philadelphia, 2004.

[23] F. Tisseur, *Tridiagonal-diagonal reduction of symmetric indefinite pairs*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 215–232.

[24] F. Tisseur and N. J. Higham, *Structured pseudospectra for polynomial eigenvalue problems, with applications*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 187–208.

[25] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[26] B. L. van der Waerden, *Modern Algebra, Vol.* 1, 2nd ed., Frederick Ungar Publishing, New York, 1953.

[27] K. Veselić, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 160–176

# A NEW ITERATIVE CRITERION FOR *H*-MATRICES*

M. ALANELLI† AND A. HADJIDIMOS†

**Abstract.** *H*-matrices appear in many areas of science and engineering, e.g., in the solution of the linear complementarity problem (LPC) in optimization theory and in the solution of large systems for real time changes of data in fluid analysis in the car industry. Classical (stationary) iterative methods used for the solution of linear systems have been shown to converge for this class of matrices. Several authors have proposed direct and iterative criteria to identify whether a certain matrix $A \in \mathbb{C}^{n,n}$ is an *H*-matrix. Based on previous and new ideas we propose a new iterative algorithm for irreducible matrices $A$ that, except in a "very special" case, decides whether $A$ is an *H*- or a non *H*-matrix. A MATLAB subroutine is given and numerical examples are provided in support of the theory developed.

**Key words.** *M*- and *H*-matrices, (generalized) strictly diagonally dominant matrices, criteria for *H*-matrices

**AMS subject classification.** 65F10

**DOI.** 10.1137/050636802

## 1. Introduction.

In numerical linear algebra, the theory of *M*- and *H*-matrices is very important for the solution of linear systems of algebraic equations by iterative methods (see, e.g., [2], [12], [24], and [26]). For example, (a) in the linear complementarity problem (LCP) [1] (see also section 10.1 of [2] for specific applications), where we are interested in finding a $z \in \mathbb{R}^n$ such that $z \geq 0$, $Mz + q \geq 0$, $z^T(Mz+q) = 0$, with $M \in \mathbb{R}^{n,n}$ and $q \in \mathbb{R}^n$ given, a sufficient condition for a solution to exist, and to be found by a modification of an iterative method, especially of SOR, is that $M$ is an *H*-matrix, with $m_{ii} > 0$, $i = 1(1)n$ [1]; (b) in fluid analysis, in the car modeling design [23], [18], it was observed that large linear systems with an *H*-matrix coefficient $A$ are solved iteratively much faster if $A$ is postmultiplied by a suitable diagonal matrix $D$, with $d_{ii} > 0$, $i = 1(1)n$, so that $AD$ is [illegible]. Although an *H*-matrix can be defined via the definition for an *M*-matrix, which will be given later on, in what follows and in the algorithms that will be presented we mostly use the following definition.

DEFINITION 1.1. [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $H$ [illegible] $D$, [illegible] $AD$ [illegible] row-wise strictly diagonally dominant

[illegible] (i) If there exists a diagonal matrix $D$ satisfying Definition 1.1 for a given $A$, there exist infinitely many $D$'s and their set is denoted by $\mathfrak{D}_A$. (ii) An *H*-matrix is also called a [illegible]. (iii) Note that Definition 1.1 for an *H*-matrix $A$ implies that $A$ is nonsingular; this is consistent with the original definition by Ostrowski [20] (see also [24]).

We remind the reader of the following definition.

DEFINITION 1.2. [illegible] $X \in \mathbb{C}^{n,n}$ [illegible] (row-wise) strictly diagonally

---

*Received by the editors July 25, 2005; accepted for publication (in revised form) by D. B. Szyld July 10, 2006; published electronically December 21, 2006.
http://www.siam.org/journals/simax/29-1/63680.html

†Department of Mathematics, University of Crete, GR-714 09 Heraklion, Greece (alanelli@math.uoc.gr, hadjidim@math.uoc.gr). The work of the first author was funded by "Herakleitos" Operational Programme for Education and Initial Vocational Training 2002–2005. Part of the work of the second author was funded by the Program Pythagoras of the Greek Ministry of Education.

dominant [illegible]

$$|x_{ii}| > \sum_{j=1,\ j\neq i}^{n} |x_{ij}|,\ i = 1(1)n. \tag{1.1}$$

In what follows we are to use for a matrix the terms [illegible] and [illegible]. For these terms we give the following definition.

DEFINITION 1.3. [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] reducible [illegible] $P \in \mathbb{R}^{n,n}$ [illegible]

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ O & A_{22} \end{bmatrix},$$

[illegible] $A_{11} \in \mathbb{C}^{r,r},\ 1 \le r \le n-1,$ [illegible] $O \in \mathbb{C}^{n-r,r}$ [illegible] not [illegible] $A$ [illegible] irreducible

Knowing that a matrix $A \in \mathbb{C}^{n,n}$ is an $H$-matrix guarantees that many iterative methods applied for the solution of the linear system $Ax = b,\ b \in \mathbb{C}^n$, possess nice convergence properties. The majority of the proposed identification criteria are iterative (see, e.g., [10], [16], [15], [17], [19], and [9]) because direct ones (see, e.g., [7], [13], [8], [6], and [4]) have high computational complexities. For example, one may argue that for this identification of $A \in \mathbb{C}^{n,n}$ a simple direct criterion would be "find the inverse of its [illegible] matrix and check if its entries are nonnegative." However, such a criterion requires $\mathcal{O}(n^3)$ floating point operations for a large dense matrix compared to $\mathcal{O}(n^2)$ per iteration for an iterative method, while for a large sparse matrix, e.g., a tridiagonal whose inverse of its [illegible], as is known, is dense, the cost is $\mathcal{O}(n^2)$ compared to $\mathcal{O}(n)$ per iteration. By the way, the only iterative criterion that takes advantage of the sparsity of $A$, as this is the case in almost all applications, is the one proposed in [9], where an extension of the compact profile technique of [14] was developed. This criterion is much cheaper compared to any of the direct or even iterative criteria.

In this work we focus on the three algorithms in [16], [17], and [19] denoted by $\mathbb{H}$, $\mathbb{L}$, and $\mathbb{B}$, respectively. In section 2, some introductory notation is given, the three algorithms are presented, and we show, by giving a number of examples, that there are cases where Algorithms $\mathbb{L}$ and $\mathbb{B}$ [illegible] converge as it was claimed. In section 3, a number of statements needed in what follows are given and some of them are proved. In section 4, by using the theory in section 3, we propose a new Jacobi-type iterative criterion for identifying $H$-matrices in pseudocode (Algorithm $\mathbb{AH}$). The convergence of the new algorithm is proved mainly for [illegible] matrices. Finally, in section 5, we implement it to a MATLAB function, present a number of numerical examples, and conclude with a couple of points.

## 2. Preliminaries, the three algorithms, and comments.

In this section we present some introductory notation, the two mathematically documented algorithms $\mathbb{H}$ in [16] and $\mathbb{L}$ in [17] and also $\mathbb{B}$ in [19], and make some comments regarding the convergence of the last two. Each of the algorithms has been slightly modified, as regards the notation, so that their similarities and differences are readily distinguished.

For the aforementioned algorithms the following matrices are defined: First, the sequence of positive diagonal matrices

$$D^{(k)},\ k = 0, 1, 2, \ldots,\quad D^{(0)} = I, \tag{2.1}$$

and also the matrices

$$A^{(k)} = A^{(k-1)}D^{(k-1)},\ k = 1, 2, 3, \ldots,\quad A^{(0)} = A, \tag{2.2}$$

for Algorithms $\mathbb{H}$ and $\mathbb{L}$, while

$$A^{(k)} = \left(D^{(k-1)}\right)^{-1} A^{(k-1)}D^{(k-1)},\ k = 1, 2, 3, \ldots,\quad A^{(0)} = (\operatorname{diag}(A))^{-1} A, \tag{2.3}$$

for Algorithm $\mathbb{B}$. From the definitions in (2.1) and (2.3), it is concluded that for Algorithm $\mathbb{B}$ there holds

$$a_{ii}^{(k)} = 1,\ i = 1(1)n,\ k = 0, 1, 2, \ldots. \tag{2.4}$$

Let $\mathbb{N} := \{1, 2, \ldots, n\}$ and

$$s_i^{(k)} = \sum_{j=1,\ j\neq i}^{n} |a_{ij}^{(k)}|,\ i = 1(1)n,\ k = 0, 1, 2, \ldots. \tag{2.5}$$

Let also

$$\mathbb{N}_1^{(k)} \equiv \mathbb{N}_1(A^{(k)}) := \left\{ i \in \mathbb{N} : |a_{ii}^{(k)}| > s_i^{(k)} \right\},$$

and $n_1^{(k)} \equiv n_1(A^{(k)})$ be its cardinality.

```
ALGORITHM ℍ.
INPUT: A matrix $A := [a_{ij}] \in \mathbb{C}^{n,n}$ and any $\varepsilon > 0$.
OUTPUT: $D = D^{(0)}D^{(1)} \cdots D^{(k)} \in \mathfrak{D}_A$ if $A$ an $H$-matrix.
1. If $\mathbb{N}_1(A) = \emptyset$ or $a_{ii} = 0$ for some $i \in \mathbb{N}$, "$A$ is an $H$-matrix," STOP; Otherwise
2. Set $A^{(0)} = A,\ D^{(0)} = I,\ k = 1$
3. Compute $A^{(k)} = A^{(k-1)}D^{(k-1)} = [a_{ij}^{(k)}]$
4. Compute $s_i^{(k)} = \sum_{j=1,\ j\neq i}^{n} |a_{ij}^{(k)}|,\ i = 1(1)n$, Update $\mathbb{N}_1^{(k)}$ and $n_1^{(k)}$
5. If $n_1^{(k)} = n$, "$A$ an $H$-matrix," STOP; Otherwise
6. Set $d = [d_i]$, where
       $d_i = \begin{cases} \frac{s_i^{(k)}+\varepsilon}{|a_{ii}^{(k)}|+\varepsilon} & \text{if } i \in \mathbb{N}_1^{(k)}, \\ 1 & \text{if } i \notin \mathbb{N}_1^{(k)} \end{cases}$
7. Set $D^{(k)} = \operatorname{diag}(d),\ k = k + 1$; Go to Step 3.
```

```
ALGORITHM 𝕃.
INPUT: A matrix $A := [a_{ij}] \in \mathbb{C}^{n,n}$ and any $\varepsilon > 0$.
OUTPUT: $D = D^{(0)}D^{(1)} \cdots D^{(k)} \in \mathfrak{D}_A$ or $\notin \mathfrak{D}_A$ if $A$ or is an $H$-matrix, respectively.
1. If $a_{ii} = 0$ for some $i \in \mathbb{N}$, "$A$ is an $H$-matrix," STOP; Otherwise
2. Set $A^{(0)} = A,\ D^{(0)} = I,\ k = 1$
3. Compute $A^{(k)} = A^{(k-1)}D^{(k-1)} = [a_{ij}^{(k)}]$
4. Compute $s_i^{(k)} = \sum_{j=1,\ j\neq i}^{n} |a_{ij}^{(k)}|,\ i = 1(1)n$, Set $n_1^{(k)} = 0$
5. If $|a_{ii}^{(k)}| > s_i^{(k)}$, $n_1^{(k)} = n_1^{(k)} + 1,\ i = 1(1)n$
6. If $n_1^{(k)} = n$, "$A$ an $H$-matrix," STOP; Otherwise
```

7. If $n_1^{(k)} = 0$, "$A$ is [illegible] an $H$-matrix," STOP; Otherwise
8. Set $d = [d_i]$, where

$$d_i \;=\; \frac{s_i^{(k)} + \varepsilon}{|a_{ii}^{(k)}| + \varepsilon},\ i = 1(1)n$$

9. Set $D^{(k)} =$ diag$(d)$, $k = k + 1$; Go to Step 3.

ALGORITHM $\mathbb{B}$.

INPUT: A matrix $A := [a_{ij}] \in \mathbb{C}^{n,n}$.
OUTPUT: $D = D^{(0)}D^{(1)} \cdots D^{(k)} \in \mathfrak{D}_{D^{-1}A} \equiv \mathfrak{D}_A$ or $\notin \mathfrak{D}_A$ if $A$ [illegible] or is [illegible] an $H$-matrix, respectively
1. If $\mathbb{N}_1(A) = \emptyset$ or $a_{ii} = 0$ for some $i \in \mathbb{N}$, "$A$ is [illegible] an $H$-matrix," STOP; Otherwise
2. Compute $s_i \;=\; \sum_{j=1,=\ j\neq i}^{n} |a_{ij}|,\ i = 1(1)n$
3. If $s_i \;= 0,\ i = 1(1)n$, "$A$ [illegible] an $H$-matrix," STOP; Otherwise
4. Set $A^{(0)} = (\text{diag}(A))^{-1} A,\ D^{(0)} = I,\ k = 1$
5. Compute $A^{(k)} \;=\; \left(D^{(k-1)}\right)^{-1} A^{(k-1)} D^{(k-1)} \;=\; [a_{ij}^{(k)}]$
6. Compute $s_i^{(k)} \;=\; \sum_{j=1,\ j\neq i}^{n} |a_{ij}^{(k)}|,\ i = 1(1)n$
7. If $s_i^{(k)} \;\leq\; 1,\ i = 1(1)n$, and $s_i^{(k)} < 1$ for at least one $i \in \mathbb{N}$, "$A$ [illegible] an $H$-matrix," STOP; Otherwise
8. If $s_i^{(k)} \;\geq\; 1,\ i = 1(1)n$, "$A$ is [illegible] an $H$-matrix," STOP; Otherwise
9. Determine $m$ such that $s_m^{(k)} \;=\; \min_{i=1(1)n} s_i^{(k)}$ for $s_i^{(k)} \neq 0$
10. Set $d = [d_i]$, where

$$d_i \;=\; \begin{cases} s_m^{(k)} & \text{if } i = m, \\ 1 & \text{if } i \neq m \end{cases}$$

11. Set $D^{(k)} = \text{diag}(d),\ k = k + 1$; Go to Step 5.

We proceed with some observations on the three algorithms.

According to the theory in [16, Theorem 2.1] or [17, Theorems 1, 2, 3, and 4] if $A$ is an $H$-matrix either Algorithm $\mathbb{H}$ or Algorithm $\mathbb{L}$ converges. On the other hand, if Algorithm $\mathbb{H}$ converges, $A$ is an $H$-matrix, while if Algorithm $\mathbb{L}$ converges, then $A$ may or may not be an $H$-matrix depending on the algorithm's output.

Regarding Algorithm $\mathbb{L}$, one can find out that in the case of a non $H$-matrix, convergence of the algorithm [illegible] always be guaranteed. When $A$ is [illegible] there may be problems as the following example shows. Let

$$A \;=\; \begin{bmatrix} 1 & -2 & 0 & -0.5 \\ -2 & 1 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \tag{2.6}$$

which is [illegible] an $H$-matrix. If one applies Algorithm $\mathbb{L}$ to $A$, where initially it is $n_1(A) = 2$, it is found that the algorithm [illegible] converge. No matter what $\varepsilon > 0$ is used, it is always $n_1(A^{(k)}) = 2,\ k = 0, 1, 2, \ldots$. This is because, for any $A^{(k)}$ strict diagonal dominance holds for its last two rows and (strict) nondiagonal dominance holds for the first two.

Let us now consider the irreducible matrices

$$A_1 \;=\; \begin{bmatrix} 1 & 0 & -0.5 \\ -0.5 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix},\quad A_2 \;=\; \begin{bmatrix} 1 & 0 & -0.5 \\ -2 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}, \tag{2.7}$$

where $A_1$ is an $H$-matrix while $A_2$ is not. In the proof of Theorem 2 in [17] it is assumed that $\varepsilon = 0$; therefore we make the same assumption here. Applying Algorithm $\mathbb{L}$, we obtain

$$A_1^{(2)} = \begin{bmatrix} 0.5 & 0 & -1 \\ -0.25 & 0.5 & 0 \\ 0 & -1 & 2 \end{bmatrix}, \quad A_2^{(2)} = \begin{bmatrix} 0.5 & 0 & -1 \\ -1 & 2 & 0 \\ 0 & -4 & 2 \end{bmatrix}, \tag{2.8}$$

$$A_1^{(3)} = \begin{bmatrix} 1 & 0 & -0.5 \\ -0.5 & 0.25 & 0 \\ 0 & -0.5 & 1 \end{bmatrix}, \quad A_2^{(3)} = \begin{bmatrix} 1 & 0 & -2 \\ -2 & 1 & 0 \\ 0 & -2 & 4 \end{bmatrix}, \tag{2.9}$$

$$A_1^{(4)} = \begin{bmatrix} 0.5 & 0 & -0.25 \\ -0.25 & 0.5 & 0 \\ 0 & -1 & 0.5 \end{bmatrix}, \quad A_2^{(4)} = \begin{bmatrix} 2 & 0 & -1 \\ -4 & 2 & 0 \\ 0 & -4 & 2 \end{bmatrix}, \tag{2.10}$$

etc. So, for the $A_1^{(k)}$ sequence we have $d_1^{(1)} = d_2^{(1)} = 0.5$, $d_3^{(1)} = 2$ and after three iterations we have again $d_1^{(4)} = d_2^{(4)} = 0.5$, $d_3^{(4)} = 2$. Obviously, this three-cyclic pattern is repeated ad infinitum. For the sequence of $A_2^{(k)}$'s a similar situation arises since then $d_1^{(1)} = 0.5$, $d_2^{(1)} = d_3^{(1)} = 2$ and $d_1^{(4)} = 0.5$, $d_2^{(4)} = d_3^{(4)} = 2$. Hence neither of the two sequences converges. This is most probably due to the assumption made in [17], that $\lim_{\varepsilon \to 0^+} \left( \lim_{k\to\infty} D^{(k)} \right) = \lim_{k\to\infty} \left( \lim_{\varepsilon\to 0^+} D^{(k)} \right)$, which may not be valid and so $\varepsilon$ [illegible] be taken to be zero. There is one more crucial point in the same theorem. It is proved that any two consecutive Jacobi matrices $B^{(k)},\ k = 0, 1, 2, \ldots$, associated with the [illegible] of the $A^{(k)}$'s (see Definition 3.2) satisfy

$$B^{(k+1)} = \left( D^{(k)} \right)^{-1} B^{(k)} D^{(k)}, \quad k = 0, 1, 2, \ldots. \tag{2.11}$$

During each iteration a similarity permutation on $D^{(k)}$ (which has to be applied also on $A^{(k)}$ and $B^{(k)}$) sets the diagonal elements $d_i^{(k)}$ in a nondecreasing order. It is then proved that for any $b_{ij}^{(k)}$ $(i < j)$, $b_{ij}^{(k)} \le b_{ij}^{(k+1)}$, while for any $b_{ij}^{(k)}$ $(i > j)$, $b_{ij}^{(k)} \ge b_{ij}^{(k+1)}$. However, in the next iteration the $d_i^{(k+1)}$'s may be in a different order as a result of which $b_{ij}^{(k+1)}$ $(i < j)$ may be found at a different position $(l, m)$, $l > m$, and the new $b_{ij}^{(k+1)}$ may not be greater than $b_{ij}^{(k)}$. Hence the sequence of $b_{ij}^{(k)}$'s may [illegible] be a monotone one as a result of which convergence [illegible] be guaranteed. For example, for $A_1$ in (2.7) it is

$$B^{(1)} = \begin{bmatrix} 0 & 0 & 0.5 \\ 0.5 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \quad \text{and} \quad D^{(1)} = \operatorname{diag}(0.5, 0.5, 2).$$

Since the elements of $D^{(1)}$ are already in a nondecreasing order, the permutation matrix to be used is $P^{(1)} = I_3$ and (2.11) gives

$$B^{(2)} = \begin{bmatrix} 0 & 0 & 2 \\ 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix} \quad \text{and} \quad D^{(2)} = \operatorname{diag}(2, 0.5, 0.5).$$

This time $P^{(2)} = [e^2 e^3 e^1]$, with $e^i,\ i = 1, 2, 3$, the columns of the identity matrix, so by permuting and using the same symbol $B^{(2)}$ for $(P^{(2)})^{-1} B^{(2)} P^{(2)}$ we have that

$B^{(2)} \equiv B^{(1)}$. This pattern repeats itself ad infinitum. Observe that the sequence of differences

$$\max_{i=1,2,3} d_i^{(k)} - \min_{i=1,2,3} d_i^{(k)} = 2 - 0.5 = 1.5, \;\; k = 1, 2, 3, \ldots,$$

[illegible] tend to zero as $k \to \infty$ as is claimed in [17]. Note, in view of (2.11), that $b_{13}^{(1)} = 0.5$ becomes $b_{13}^{(2)} = 2$ but after the similarity permutation takes place the new $b_{13}^{(2)}$ is again 0.5.

So, even in the case of irreducible matrices, for Algorithm $\mathbb{L}$ to be valid, a proof of Theorem 2 in [17] has to be provided in which the quantity $\varepsilon > 0$ should be used throughout the proof, and this is because the value $\varepsilon = 0$ leads to erroneous conclusions.

Let us now come to Algorithm $\mathbb{B}$ in [19]. Besides the conclusion which follows the restriction in Step 7 which holds if and only if $A$ is irreducible, one can make similar observations as in the case of Algorithm $\mathbb{L}$. Consider the following two matrices, where the first one is reducible and the second irreducible.

$$A_3 = \begin{bmatrix} 1 & -2 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 0 & -\frac{1}{4} & 1 & -\frac{1}{2} \\ -\frac{1}{4} & 0 & -\frac{1}{2} & 1 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 1 & -2 & -1 & 0 \\ -2 & 1 & 0 & -1 \\ 0 & -\frac{1}{4} & 1 & -\frac{1}{2} \\ -\frac{1}{4} & 0 & -\frac{1}{2} & 1 \end{bmatrix}. \tag{2.12}$$

For both matrices $s_3^{(1)} = s_4^{(1)} = \min_{i=1(1)4} s_i^{(1)} = \frac{3}{4}$; let us take $m = 3$. After the first iteration we have

$$A_3^{(2)} = \begin{bmatrix} 1 & -2 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 0 & -\frac{1}{3} & 1 & -\frac{2}{3} \\ -\frac{1}{2} & 0 & -\frac{3}{8} & 2 \end{bmatrix}, \quad A_4^{(2)} = \begin{bmatrix} 1 & -2 & -\frac{3}{4} & 0 \\ -2 & 1 & 0 & -1 \\ 0 & -\frac{1}{3} & 1 & -\frac{2}{3} \\ -\frac{1}{4} & 0 & -\frac{3}{8} & 1 \end{bmatrix}. \tag{2.13}$$

For the second iteration $s_m^{(2)} = s_4^{(2)} = \frac{5}{8}$ and therefore

$$A_3^{(3)} = \begin{bmatrix} 1 & -2 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ 0 & -\frac{1}{3} & 1 & -\frac{5}{12} \\ -\frac{2}{5} & 0 & -\frac{3}{5} & 1 \end{bmatrix}, \quad A_4^{(3)} = \begin{bmatrix} 1 & -2 & -\frac{3}{4} & 0 \\ -2 & 1 & 0 & -\frac{5}{8} \\ 0 & -\frac{1}{3} & 1 & -\frac{5}{12} \\ -\frac{2}{5} & 0 & -\frac{3}{5} & 1 \end{bmatrix}, \tag{2.14}$$

and then the sequence of $s_m$'s is $s_3^{(3)} = \frac{3}{4}$, $s_4^{(4)} = \frac{17}{20}$, $s_3^{(5)} = \frac{33}{36}$, and so on. Obviously, for either example, the algorithm does [illegible] converge.

Based on all previous examples it is realized that one must distinguish the reducible from the irreducible case. However, because in the case of a large matrix with an unstructured pattern of nonzero elements checking for irreducibility may be of prohibitive computational complexity, we leave aside this issue.

**3. Preliminaries and background material.** For the analysis that will follow, we have to recall some definitions and give a number of useful statements, most of which can be found in [2], [12], and [24].

DEFINITION 3.1. [illegible] $A \in \mathbb{R}^{n,n}$ [illegible] $M$ matrix [illegible] $A = sI - B$, [illegible] $B \geq 0$ [illegible] $\rho(B) < s$, [illegible] $\rho(.)$ [illegible]

[illegible] In Definition 3.1 and in this context an $M$-matrix is assumed to be nonsingular.

Lemma 3.1. [illegible] $A \in \mathbb{R}^{n,n}$ [illegible] $M$ [illegible] $PAP^T$ [illegible] $P$ [illegible]

Definition 3.2. [illegible] comparison [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $\mathcal{M}(A)$ [illegible]

$$m_{ij} = \begin{cases} |a_{ii}| & \text{[illegible]}\ i = j = 1(1)n, \\ -|a_{ij}| & \text{[illegible]}\ i,j = 1(1)n,\ i \neq j. \end{cases}$$

Lemma 3.2. [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $H$ [illegible] $M$ [illegible]

Lemma 3.3. [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $H$ [illegible]

[illegible] Lemmas 3.2 and 3.3 can be used as alternative and equivalent definitions to Definition 1.1 for an $H$-matrix.

Lemma 3.4. [illegible] $A \in \mathbb{C}^{n,n}$, [illegible] $a_{ii} \neq 0$, $i = 1(1)n$, [illegible] $B = EA$ [illegible] $E = \mathrm{diag}(e_1, e_2, \ldots, e_n) \in \mathbb{C}^{n,n}$ [illegible] $J_A$ [illegible] $J_B$ [illegible] $A$ [illegible] $B$ [illegible] $J_A$ [illegible] $J_B$ [illegible]

Based on the last two lemmas, our objective would be that of approximating the spectral radius of the Jacobi iteration matrix $B = J_{\mathcal{M}(A)}$ associated with the comparison matrix of a given $A \in \mathbb{C}^{n,n}$. In fact we will be able to prove that if $A$ is irreducible and $\rho(B) < 1$ ($A$ [illegible] an $H$-matrix) or $\rho(B) > 1$ ($A$ is [illegible] an $H$-matrix), there exists an algorithm that converges. Also, the same algorithm converges in case $A$ [illegible] a reducible $H$-matrix. In case $A$ is [illegible] an $H$-matrix the algorithm may [illegible] converge.

The algorithm we are to propose is based on a modification of the well-known [illegible] (see, e.g., [25] and, more specifically, [5] and [11]), which all iterative criteria for $H$-matrices use indirectly as a starting point. The [illegible] is stated below.

Theorem 3.1 (the Power Method). [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $\lambda_i$, $i = 1(1)n$ [illegible]

$$|\lambda_1| > |\lambda_j|,\ j = 2(1)n.$$

[illegible]

$$x^{(k)} = Ax^{(k-1)},\ k = 1, 2, 3, \ldots,\ \text{[illegible]}\ x^{(0)} \in \mathbb{C}^n \backslash \{0\}. \tag{3.1}$$

[illegible] $x^{(0)}$ [illegible] $\lambda_1$ [illegible]

$$\lambda_1 = \lim_{k \to \infty} \frac{(Ax^{(k)})_i}{x_i^{(k)}} \quad \text{[illegible]}\ x_i^{(k)} \neq 0, \quad i = 1(1)n. \tag{3.2}$$

Since, from now on, we are to deal with nonnegative irreducible matrices we give a number of statements that will be used later on and can be found in either [2] or [24].

Theorem 3.2 (see [24]). [illegible] $A \in \mathbb{R}^{n,n}$ $A \geq 0$ [illegible] $\rho(A)$ [illegible]

Definition 3.3 (see [24]). [illegible] $A \in \mathbb{R}^{n,n}$ $A \geq 0$ [illegible] $k$ [illegible] $\rho(A)$ [illegible] $k = 1$ [illegible] $A$ [illegible] primitive [illegible] $k > 1$ [illegible] $A$ [illegible] cyclic of index $k$

Lemma 3.5 (see [2]). [illegible] $A \in \mathbb{R}^{n,n}$ $A \geq 0$ [illegible] $\sum_{i=1}^{n} a_{ii} > 0$ [illegible]

Theorem 3.3 (see [24]). [illegible] $A \in \mathbb{R}^{n,n}$, $A \geq 0$ [illegible] $P^*$ [illegible] hyperoctant [illegible] $x > 0$ [illegible] $x \in P^*$ [illegible]

$$\min_{i=1(1)n} \left\{ \frac{\sum_{j=1}^{n} a_{ij} x_j}{x_i} \right\} < \rho(A) < \max_{i=1(1)n} \left\{ \frac{\sum_{j=1}^{n} a_{ij} x_j}{x_i} \right\}$$

[illegible]

$$\frac{\sum_{j=1}^{n} a_{ij} x_j}{x_i} = \rho(A), \ i = 1(1)n.$$

[illegible] 3.1. A fixed vector $x \in P^*$ in Theorem 3.3 can be considered as representing all of its positive multiples ($cx$, $c \in \mathbb{R}_+$). Obviously, for any positive multiple $cx$, the theorem holds true and the three ratios in it remain unchanged.

Below we prove a key statement needed for the application of the Power Method (3.1) to an irreducible, nonnegative, primitive matrix $A \in \mathbb{R}^{n,n}$. In passing we mention that for reducible, nonnegative, and primitive matrices, an issue outside this work and on which we have been working, references [3], [21], and [22] may be of great help since they refer, among other issues, to the eigenspaces of nonnegative matrices, the nonnegative Jordan bases, and the orthogonality of some eigenspaces involved.

Theorem 3.4. [illegible] $A \in \mathbb{R}^{n,n}$, $A \geq 0$ [illegible] $\lambda_1 = \rho(A)$ [illegible] $A = SJS^{-1}$ [illegible] $J = \operatorname{diag}(J_1, J_2, \ldots, J_p)$, $J_i \in \mathbb{C}^{n_i,n_i}$, $i = 1(1)p$, $\sum_{i=1}^{p} n_i = n$, [illegible] $S = [s_1\, s_2\, s_3 \, \ldots \, s_n]$ [illegible] $A$ [illegible] $x \in P^*$, [illegible] $s_i$, $i = 1(1)n$, [illegible] $s_1$ [illegible] $\lambda_1$

[illegible] Due to its properties, $A$ has a unique positive eigenvalue $\lambda_1$ ([illegible]) equal to its spectral radius and a positive eigenvector $s_1$ ([illegible]) associated with it. Let $x \in P^*$ be written as $x = \sum_{i=1}^{n} \eta_i s_i$. Since $A^T$ and $A$ have identical eigenvalue spectra, and $A^T$ is also nonnegative, irreducible, and primitive, its [illegible] is $\lambda_1$; let $z\,(>0)$ be its [illegible]. Consequently,

$$0 \ < \ z^T x \ = \ \sum_{i=1}^{n} \eta_i z^T s_i. \tag{3.3}$$

Noting that $z$ is also the left eigenvector of $A$ corresponding to $\lambda_1$, consider any of the Jordan blocks $J_m \in \mathbb{C}^{n_m,n_m}$, $m = 2(1)p$, of $J$ and let $m = 2$ for the sake of convenience. From the Jordan canonical form of $A$ we have $AS = SJ$ and so

$$As_1 \ = \ \lambda_1 s_1, \ As_2 \ = \ \lambda_2 s_2, \ \text{and } As_i \ = \ s_{i-1} + \lambda_2 s_i, \ i = 3(1)n_2 + 1. \tag{3.4}$$

Recalling that $z$ is a left eigenvector and $s_2$ is a (right) eigenvector of $A$ corresponding to different eigenvalues $\lambda_1$ and $\lambda_2$, respectively, there will hold that $z^T s_2 = 0$ (see, e.g., Theorem 1.1.7 of [11]). From the third set of relations in (3.4) we have for $i = 3$, $z^T A s_3 = z^T(s_2 + \lambda_2 s_3)$, from which $z^T A s_3 - \lambda_2 z^T s_3 = 0$ or $(\lambda_1 - \lambda_2) z^T s_3 = 0$ and $z^T s_3 = 0$. Inductively, it is found that $z^T s_i = 0$, $i = 3(1)n_2 + 1$. What we have just proved for $J_2$ applies to every Jordan block $J_m$, $m = 2(1)p$; consequently

$$z^T s_i \ = \ 0, \ \ i = 2(1)n. \tag{3.5}$$

In view of (3.5), (3.3) becomes $z^T x = \eta_1 z^T s_1 > 0$; hence the positivity of $z$, $x$, and $s_1$ implies directly that $\eta_1 > 0$ and the theorem has been proved. □

In the next section, using some ideas from Algorithms $\mathbb{H}$, $\mathbb{L}$, and $\mathbb{B}$, as well as some new ones, Definition 1.1, or one of its equivalents in Lemmas 3.2 and 3.3, we mainly exploit Theorems 3.1, 3.3, and 3.4 and propose a new algorithm (Algorithm $\mathbb{AH}$) which, in case of an [illegible] matrix $A \in \mathbb{C}^{n,n}$, converges.

For our algorithm to identify if an irreducible matrix $A \in \mathbb{C}^{n,n}$, with $a_{ii} \neq 0$, $i = 1(1)n$, is an $H$-matrix or not we apply the Power Method (3.1) to the irreducible and primitive nonnegative matrix $|A^{(0)}|$, where $A^{(0)} = (\text{diag}(A))^{-1}A$, with $x^{(0)} = e (\in P^*)$ and $e$ the vector of ones. Considering $|A^{(k)}| = I + B^{(k)}$, $k = 0, 1, 2, \ldots$, and noting that $B^{(0)}$ is the Jacobi matrix associated with the comparison matrix of $A$, $J_{\mathcal{M}(A)}$, we stop iterating as soon as one of the following three possibilities occurs: all the components of the vector $(I - B^{(k)})e$ are positive, negative, or zero. It is understood that the similar to $A^{(0)}$ matrix $A^{(k)}$ is given by $A^{(k)} = (\text{diag}(x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)}))^{-1} A^{(k-1)} \text{diag}(x_1^{(k-1)}, x_2^{(k-1)}, \ldots, x_n^{(k-1)})$, with $x^{(0)} = e$. Then, according to Theorem 3.3, $[\min_{i=1(1)n} s_i^{(k)}, \max_{i=1(1)n} s_i^{(k)}]$, $s_i^{(k)} = \sum_{j=1(1)n} |b_{ij}^{(k)}|$, $i = 1(1)n$, is an interval in which the [illegible] $\rho(B^{(0)}) = \rho(|A^{(0)}|) - 1$ of $J_{\mathcal{M}(A)}$ lies and also the vector $\text{diag}(x^{(0)})\text{diag}(x^{(1)}) \cdots \text{diag}(x^{(k)})e$ is an approximation to the [illegible] of $|A^{(0)}|$ or of $B^{(0)}$. As is clear, it is [illegible] necessary to go through all the iterations of the Power Method (3.1) since it suffices to go as far as one of the aforementioned stopping criteria is satisfied. Hence, our algorithm will converge at a much earlier stage than the Power Method.

**4. The new algorithm.** In our algorithm we follow Algorithms $\mathbb{L}$, [illegible] the need of a parameter $\varepsilon$ in the definition of the $d_i$'s, and adopt a more general normalization than that of Algorithm $\mathbb{B}$ so that at each stage $A^{(k)}$, $k = 0, 1, 2, \ldots$, has diagonal elements $a_{ii}^{(k)} = 1$, $i = 1(1)n$, $k = 0, 1, 2, \ldots$. The proofs of our main claims will be given analytically after the presentation of the new algorithm below.

ALGORITHM $\mathbb{AH}$.

```
INPUT: An irreducible matrix $A := [a_{ij}] \in \mathbb{C}^{n,n}$.
OUTPUT: $D = D^{(0)} D^{(1)} \cdots D^{(k)} \in \mathfrak{D}_{D^{-1}A} \equiv \mathfrak{D}_A$ or $\notin \mathfrak{D}_A$ if $A$ [illegible] or is [illegible] an $H$-matrix, respectively
1. If $a_{ii} = 0$ for some $i \in \mathbb{N}$, "$A$ is [illegible] an $H$-matrix," STOP; Otherwise
2. Set $D = I$, $A^{(0)} = (\text{diag}(A))^{-1} A$, $D^{(0)} = I$, $k = 1$
3. Compute $D = DD^{(k-1)}$, $A^{(k)} = \left(D^{(k-1)}\right)^{-1} A^{(k-1)} D^{(k-1)} = [a_{ij}^{(k)}]$
4. Compute $s_i^{(k)} = \sum_{j=1,\ j\neq i}^{n} |a_{ij}^{(k)}|$, $i = 1(1)n$, $s^{(k)} = \min_{i=1(1)n} s_i^{(k)}$, $S^{(k)} = \max_{i=1(1)n} s_i^{(k)}$
5. If $s^{(k)} > 1$, "$A$ is [illegible] an $H$-matrix," STOP; Otherwise
6. If $S^{(k)} < 1$, "$A$ [illegible] an $H$-matrix," STOP; Otherwise
7. If $S^{(k)} = s^{(k)}$, "$\mathcal{M}(A)$ is [illegible]," STOP; Otherwise
8. Set $d = [d_i]$, where

       $d_i = \frac{1 + s_i^{(k)}}{1 + S^{(k)}}, \ i = 1(1)n$

9. Set $D^{(k)} = \text{diag}(d)$, $k = k + 1$; Go to Step 3.
```

THEOREM 4.1. [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $\mathbb{AH}$ [illegible] $\det(\mathcal{M}(A)) = 0$

[illegible]. If any $a_{ii}$, $i = 1(1)n$, is zero the algorithm terminates (Step 1). Assuming $a_{ii} \neq 0$, $i = 1(1)n$, then by Lemma 3.4, $A^{(0)}$ and $A$ have the same Jacobi iteration matrices. Therefore, $A^{(0)}$ is an $H$-matrix if and only if $A$ is (Step 2). Since $A$ is irreducible so are $A^{(0)}$ and $|A^{(0)}|$. Also, by Lemma 3.5, $|A^{(0)}|$ is primitive. At the $k$th iteration step we form $A^{(k)}$ and indirectly the comparison matrix $\mathcal{M}(A^{(k)})$. Note that the Jacobi iteration matrix $B^{(k)} = |A^{(k)}| - I$, associated with $\mathcal{M}(A^{(k)})$, will be

$$J_{\mathcal{M}(A^{(k)})} = B^{(k)} = I - \mathcal{M}(A^{(k)}) = \begin{bmatrix} 0 & |a_{12}^{(k)}| & \dots & |a_{1n}^{(k)}| \\ |a_{21}^{(k)}| & 0 & \dots & |a_{2n}^{(k)}| \\ \vdots & \vdots & \ddots & \vdots \\ |a_{n1}^{(k)}| & |a_{n2}^{(k)}| & \dots & 0 \end{bmatrix} = [b_{ij}^{(k)}]. \tag{4.1}$$

Because of the similarity transformation (Step 3), we have by induction that $|A^{(k)}| = I + B^{(k)}$ is nonnegative, irreducible, and primitive. Observe that (Step 4)

$$s_i^{(k)} = \sum_{j=1}^{n} b_{ij}^{(k)} = \sum_{j=1,\, j\neq i}^{n} |a_{ij}^{(k)}|, \ i = 1(1)n,$$

are the row sums of the off-diagonal elements of $-\mathcal{M}(A^{(k)})$. Furthermore, $B^{(k)} = \left(D^{(k-1)}\right)^{-1} B^{(k-1)} D^{(k-1)}$, so $B^{(k)}$ and $B^{(k-1)}$ are similar and inductively so are $B^{(k)}$ and $B^{(0)}$. Hence all $B^{(k)}$'s, $k = 0, 1, 2, \dots$, have the same eigenvalue spectra and the same spectral radii. If any of the criteria in Steps 5–7 are satisfied, then according to the discussion in the last paragraph of the previous section the algorithm terminates. By choosing $d^{(0)} = e \in P^*$, we define

$$d^{(k)} = |A^{(0)}| d^{(k-1)} = \cdots = |A^{(0)}|^k d^{(0)}, \ k = 1, 2, \dots,$$

and by induction we have $d^{(k)} \in P^*$. By the definition of $D$ and $D^{(k)}$'s in the algorithm (Steps 8–9, 2), $d^{(k)} = [d_1 \ d_2 \ \dots \ d_n]^T$, where the $d_i$'s refer to the current matrix $D$ after Step 2 of the algorithm is executed, is an approximation to the [illegible] of $|A^{(0)}|$; the presence of the denominators in Step 8 aims at the avoidance of the uncontrollable increase of the components of $d^{(k)}$. Forming now the ratios $\frac{d_i^{(k)}}{d_i^{(k-1)}}$, $i = 1(1)n$, where $d_i^{(k-1)} \neq 0$, $i = 1(1)n$, $\forall\, k = 1, 2, \dots$, since $d^{(k-1)} \in P^*$, by (4.1) we obtain that

$$\frac{d_i^{(k)}}{d_i^{(k-1)}} = \frac{\left(|A^{(k-1)}| d^{(k-1)}\right)_i}{d_i^{(k-1)}} = \frac{\sum_{j=1}^n |a_{ij}^{(k-1)}|(1 + s_j^{(k-1)})}{1 + s_i^{(k-1)}} = 1 + s_i^{(k)}, \tag{4.2}$$

$$\text{with } s_i^{(0)} = 0, \ i = 1(1)n.$$

By virtue of Theorem 3.4, the requirement of having a nonzero component of $d^{(0)}$ $(> 0)$ along the Perron vector of $|A^{(0)}|$ is satisfied. Since then all the assumptions of Theorem 3.1 hold, there will be $\lim_{k\to\infty} s_i^{(k)} = \rho(B^{(0)})$, $i = 1(1)n$. Therefore, by Theorem 3.3 (and also by Lemma 4.1 and Theorem 4.3 below) we have that

$$1 + s^{(1)} \leq 1 + s^{(2)} \leq \cdots \leq 1 + \rho(B^{(0)}) \leq \cdots \leq 1 + S^{(2)} \leq 1 + S^{(1)}, \tag{4.3}$$

where $s^{(k)} = \min_{i=1(1)n} s_i^{(k)}$ and $S^{(k)} = \max_{i=1(1)n} s_i^{(k)}$, and that

$$\lim_{k\to\infty} s^{(k)} = \lim_{k\to\infty} S^{(k)} = \rho(B^{(0)}).$$

All elements the of the two monotonically convergent sequences $s^{(k)}$ and $S^{(k)}$, $k = 1, 2, \dots$ (nondecreasing and nonincreasing, respectively), except for a finite number of them, will belong to $\big(\rho(B^{(0)}) - \varepsilon, \rho(B^{(0)}) + \varepsilon\big)$ for every $\varepsilon > 0$. This, in turn, guarantees convergence of the algorithm. To prove our claim we have to distinguish the following three cases.

(i) If $\rho(B^{(0)}) < 1$, then setting $\varepsilon = 1 - \rho(B^{(0)})$, all the terms of $S^{(k)}$, $k = 1, 2, \dots$, but a finite number of them belong to $(2\rho(B^{(0)}) - 1, 1)$. Consequently, there will exist an integer $k_S$ such that for all $k \geq k_S$ it will be $S^{(k)} < 1$, and so by Lemma 3.3 $A$ [illegible] an $H$-matrix.

(ii) If $\rho(B^{(0)}) > 1$, we set $\varepsilon = \rho(B^{(0)}) - 1$; hence analogously only a finite number of the terms of $s^{(k)}$, $k = 1, 2, \dots$, will lie outside $(1, 2\rho(B^{(0)}) - 1)$, and by Lemma 3.3, as before, $A$ is [illegible] an $H$-matrix.

(iii) If $\rho(B^{(0)}) = 1$, the algorithm, theoretically, may converge or not. Computationally, it is inconclusive. □

The converse of Theorem 4.1 is easy to prove. More specifically, we have the following theorem.

THEOREM 4.2. [illegible] $A \in \mathbb{C}^{n,n}$ [illegible] $\mathbb{AH}$ [illegible]

[illegible]. Assume that the algorithm converges. Then, we have to distinguish cases based on the algorithm's output and on the step from which the algorithm exited.

(i) Suppose that the output is "$A$ [illegible] an $H$-matrix." The only possible exit is from Step 6. In such a case, $S^{(k)} < 1$. By Theorem 3.3, there holds

$$s^{(k)} \leq \rho(B^{(0)}) \leq S^{(k)} < 1 \Longrightarrow \rho(B^{(0)}) < 1.$$

Thus by Lemma 3.3, $A^{(k)}$ is an $H$-matrix and so are $A^{(0)}$ and $A$.

(ii) Suppose that the output is "$A$ is [illegible] an $H$-matrix." Possible exits are from Steps 1 and 5.

(a) If the exit is from Step 1, then $a_{ii} = 0$ for some $i \in \mathbb{N}$, meaning that $A$ is [illegible] an $H$-matrix.

(b) If the exit is from Step 5, then by an analogous argument to that in (i) previously, there holds $\rho(B^{(0)}) > 1$; hence $A$ is [illegible] an $H$-matrix.

(iii) If the exit is from Step 7, then since the algorithm has just passed from Steps 5 and 6 it is implied that $s^{(k)} \leq 1$ and $S^{(k)} \geq 1$, respectively. Therefore $s^{(k)} = S^{(k)} = 1$; hence, by Theorem 3.3, $\rho(B^{(k)}) = 1$ and as a consequence of the matrix similarities we have from Lemmas 3.1 and 3.2 (or 3.3) that $\rho(B^{(0)}) = 1$. By virtue of Lemma 3.4 it follows that $A$ is [illegible] an $H$-matrix. If the aforementioned equality in Step 7 has been produced computationally, then $\rho(B^{(0)})$ may be very close to unity and therefore [illegible] conclusion whether $A$ is an $H$-matrix or not can be drawn. □

[illegible] 4.1. Practically, each iteration of Algorithm $\mathbb{AH}$, excluding initializations, substitutions, and comparisons, consists of the following three major steps which show the total cost per iteration step.

1. Compute $I + B^{(k)} = (\mathrm{diag}(d_1^{(k-1)}, \dots, d_n^{(k-1)}))^{-1}(I + B^{(k-1)})\mathrm{diag}(d_1^{(k-1)}, \dots, d_n^{(k-1)})$.

2. Compute $s_i^{(k)} = \frac{\left(B^{(k-1)}d^{(k-1)}\right)_i}{d_i^{(k-1)}}, \quad i = 1(1)n$.

3. Compute $d_i^{(k)} = \frac{1+s_i^{(k)}}{1+S^{(k)}}, \quad i = 1(1)n$.

To conclude this section we present a theorem based on the assumptions and notation of Theorems 4.1 and 4.2 which guarantees that after at most $l$ iterations $(l \leq [\frac{n}{2}])$ it will be $[s^{(k+l)}, S^{(k+l)}] \subset [s^{(k)}, S^{(k)}]$ while after at most $m$ iterations

$(l \leq m \leq n-1)$ there will hold $s^{(k)} < s^{(k+m)} \leq \rho(B^{(0)}) \leq S^{(k+m)} < S^{(k)}$. This theorem is based on the following lemma.

LEMMA 4.1. [illegible] 4.1 [illegible]

$$s^{(k)} \leq s_i^{(k+1)} = \frac{\sum_{j=1}^{n} b_{ij}^{(k)}(1+s_j^{(k)})}{1+s_i^{(k)}} \leq S^{(k)},\ i = 1(1)n, \tag{4.4}$$

$a_{ii} \neq 0$, $i = 1(1)n$ [illegible] $S^{(k)}$, [illegible] $s^{(k)}$ [illegible]

[illegible]. The first part of the lemma is an immediate consequence of relationships (4.2) and (4.3), where at least one of the two inequalities is strict for otherwise the algorithm would terminate. For the second part suppose that there are $n_1^{(k)} (< n)$ rows of $B^{(k)}$ that have the maximum row sum $S^{(k)}$, $n_3^{(k)}$ $(\leq n - n_1^{(k)})$ rows that have the minimum row sum $s^{(k)}$ and that for the remaining $n_2^{(k)}$ $(= n - n_1^{(k)} - n_3^{(k)} \geq 0)$ rows, the above inequalities are strict. Suppose also that by a similarity permutation with permutation matrix $P^{(k)}$, we bring the aforementioned $n_1^{(k)}$ rows first, the $n_2^{(k)}$ rows next, and the $n_3^{(k)}$ rows last, by keeping the same symbol $B^{(k)}$ for $P^{(k)}B^{(k)}(P^{(k)})^T$. So, the following three series of relationships hold:

$$\begin{aligned} S^{(k)} = \max\nolimits_{i=1(1)n_1^{(k)}} s_i^{(k)},\ \ s^{(k)} < s_i^{(k)} < S^{(k)},\ i = n_1^{(k)} + 1(1)n_1^{(k)} + n_2^{(k)}, \\ s^{(k)} = \min\nolimits_{i=n_1^{(k)}+n_2^{(k)}+1(1)n} s_i^{(k)}, \end{aligned} \tag{4.5}$$

and, in agreement with (4.5), $B^{(k)}$ will have the block partitioned form

$$B^{(k)} = \left[\begin{array}{c|c|c} B_{11}^{(k)} & B_{12}^{(k)} & B_{13}^{(k)} \\ \hline B_{21}^{(k)} & B_{22}^{(k)} & B_{23}^{(k)} \\ \hline B_{31}^{(k)} & B_{32}^{(k)} & B_{33}^{(k)} \end{array}\right]. \tag{4.6}$$

From the irreducibility of $A$, and therefore of $B^{(k)}$, submatrix $\left[\ B_{12}^{(k)}\ \middle|\ B_{13}^{(k)}\ \right]$ will have at least one nonzero element $b_{ij}^{(k)}$, for otherwise $B^{(k)}$ would be reducible. The same is true for submatrix $\left[\ B_{31}^{(k)}\ \middle|\ B_{32}^{(k)}\ \right]$ and also for at least one of $B_{21}^{(k)}$ and $B_{23}^{(k)}$. As we have already seen $B^{(k+1)} = (D^{(k)})^{-1}B^{(k)}D^{(k)}$; therefore, by summing up all the elements in each row of $I + B^{(k+1)}$, one can find out that for at least one row $i \in \{1, 2, \ldots, n_1^{(k)}\}$ there will hold

$$1 + s^{(k)} < 1 + s_i^{(k+1)} = 1 + \sum_{j=1}^{n} b_{ij}^{(k+1)} < 1 + S^{(k)}. \tag{4.7}$$

To prove this, suppose that $b_{i_1 j_1}^{(k)} \neq 0$ for some $i_1 \in \{1, \ldots, n_1^{(k)}\}$ and $j_1 \in \{n_1^{(k)} + 1, \ldots, n\}$; then we may see that the second inequality of (4.7) is proved as follows:

$$\begin{aligned} s_{i_1}^{(k+1)} = \frac{\sum_{j=1,\, j\neq i_1}^{n} b_{i_1 j}^{(k)}(1+s_j^{(k)})}{1+s_{i_1}^{(k)}} < S^{(k)} \iff \sum_{j=1,\, j\neq i_1}^{n} b_{i_1 j}^{(k)} + \sum_{j=1,\, j\neq i_1}^{n} b_{i_1 j}^{(k)} s_j^{(k)} \\ < S^{(k)} + s_{i_1}^{(k)} S^{(k)} \iff \end{aligned}$$

$$\sum_{j=1,\, j\neq i_1}^{n} b_{i_1 j}^{(k)} s_j^{(k)} < s_{i_1}^{(k)} S^{(k)} \ (s_{i_1}^{(k)} = S^{(k)}) \Longleftrightarrow \sum_{j=1,\, j\neq i_1}^{n} b_{i_1 j}^{(k)} s_j^{(k)} < \sum_{j=1,\, j\neq i_1}^{n} b_{i_1 j}^{(k)} S^{(k)}.$$

Obviously, $s_j^{(k)} \leq S^{(k)},\ j = 1(1)n,\, j \neq i_1$. But since $b_{i_1 j_1}^{(k)} \neq 0$, $b_{i_1 j_1} s_{j_1}^{(k)} < b_{i_1 j_1} S^{(k)}$, because $s_{j_1}^{(k)} < S^{(k)}$, $j_1 = n_1^{(k)} + 1(1)n$. Therefore the last strict inequality in the equivalences is true and so is the very first one. In an analogous way it is proved that the minimum row sum in the last $n_3^{(k)}$ rows is increased in at least one row. Finally, in a similar way, it can be proved that any of the new row sums in $B^{(k+1)}$, $i = n_1^{(k)} + 1(1)n_1^{(k)} + n_2^{(k)}$, can become neither as big as $S^{(k)}$ nor as small as $s^{(k)}$. ☐

THEOREM 4.3. [illegible] 4.1 [illegible] 4.1 [illegible] $k$ [illegible] $l(\leq [\frac{n}{2}])$ [illegible] $m(l \leq m \leq n-1)$ [illegible]

$$[s^{(k+q)}, S^{(k+q)}] \subset [s^{(k)}, S^{(k)}], \quad q = l, m; \tag{4.8}$$

[illegible] $q = m$ [illegible]

$$s^{(k)} < s^{(k+m)} \leq \rho(B^{(0)}) \leq S^{(k+m)} < S^{(k)}. \tag{4.9}$$

[illegible]. For the inclusion (4.8) to hold for $q = l$, and hence for $q = m$, there must be either $s^{(k)} < s^{(k+l)}$ for the minimum row sum or $S^{(k+l)} < S^{(k)}$ for the maximum row sum. Using the notation of Lemma 4.1, it is obvious that the "worst" case we could have is $n_2^{(k)} = 0$, and (i) for $n$ even $n_1^{(k)} = n_3^{(k)} = \frac{n}{2}$, while (ii) for $n$ odd either $n_1^{(k)} = \frac{n-1}{2}$ and $n_3^{(k)} = \frac{n+1}{2}$ or $n_1^{(k)} = \frac{n+1}{2}$ and $n_3^{(k)} = \frac{n-1}{2}$. Lemma 4.1 implies that in either case, $n$ even or odd, the maximum number of iterations needed to have one of the two inequalities strict is $l = [\frac{n}{2}]$. Finally, for the set of strict inequalities in (4.9) to be satisfied the "worst" case is again to have $n_2^{(k)} = 0$ and either $n_1^{(k)} = n - 1$ or $n_3^{(k)} = n - 1$. So, combining it with the result of Lemma 4.1 we have that inequalities (4.9) are both satisfied after at most $m = n - 1$ iterations. ☐

**5. A MATLAB function, examples, and comments.** We begin this section by giving a MATLAB function that implements our $\mathbb{AH}$ Algorithm. The MATLAB function has as a guide Algorithm $\mathbb{H}$ of [16].

```
function [s_min, s_max, k, dd]=ahalgo(n, a, maxit)
% INPUT: n = dimension of a square (complex) matrix,
%        a = an n-by-n (complex) matrix,
%        maxit = maximum number of iterations allowed
% OUTPUT: dd = diagonal matrix D (if "A IS an H-matrix" or if "A is NOT an
H-matrix"),
%             = [] (if "A is NOT an H-matrix; it has at least one zero diagonal
element"
%             or if "M(A) IS SINGULAR"),
%        s_min = smallest row sum of moduli of the Jacobi matrix of M(A(k)),
%        s_max = largest row sum of moduli of the Jacobi matrix of M(A(k)),
%        k = number of iterations performed
finish=0; k=1; dd=eye(n);
if (1-all(diag(a)))
    "A is NOT an H-matrix; It has at least one zero diagonal element"
    finish=1; s_min=0; s_max=Inf; k=k-1; dd=[];
```

```
end
if (finish == 0)
    for i=1:n
        a(i,1:n)=abs(a(i,1:n));
    end
    a=inv(diag(diag(a)))*a;
    for i=1:n
        a(i,i)=1;
    end
end
while (finish == 0 & k < maxit+1)
    for i=1:n
        s(i)=sum(a(i,1:n))-1;
    end
    s_min=min(s); s_max=max(s);
    if s_min > 1
        "s_min > 1, A is NOT an H-matrix"
        finish=1;
        break
    elseif s_max < 1
        "s_max < 1, A IS an H-matrix"
        finish=1;
        break
    elseif (s_min==s_max)
        "s_min=s_max, M(A) (to the MATLAB precision) IS SINGULAR"
        finish=1;
        break
    else
        for i=1:n
            d(i)=(1+s(i))/(1+s_max);
        end
    end
    k=k+1; diagonal=diag(d);
    dd=dd*diagonal; d_1=inv(diagonal); a=d_1*a*diagonal;
    for i=1:n
        a(i,i)=1;
    end
end
if (k==maxit+1 & finish==0)
    k=k-1; dd=[];
    "Inconclusive; increase maxiter"
end
% end of the function ahalgo
```

A number of numerical examples, where the matrix considered is irreducible, run with the given MATLAB function are presented.

First we consider the irreducible matrices that played the roles of counterexamples in section 2 for which Algorithm $\mathbb{L}$, with $\varepsilon = 0$, enters a loop and thus fails to converge.

1. $A_1$ in (2.7):

: "$A$ is an $H$-matrix," $\max_{i=1(1)3} s_i^{(4)} = 0.8750 < 1$, $D = \operatorname{diag}(0.5000, 0.3125, 0.8750)$.

2. $A_2$ in (2.7):
: "$A$ is NOT an $H$-matrix," $\min_{i=1(1)3} s_i^{(4)} = 1.1429 > 1$,
$D = \text{diag}(0.3333,\ 0.5333,\ 0.9333)$.

Next, the counterexample in (2.12), where Algorithm $\mathbb{B}$ fails to converge, is presented.

3. $A_4$ in (2.12):
: "$A$ is NOT an $H$-matrix," $\min_{i=1(1)4} s_i^{(2)} = 1.0714 > 1$,
$D = \text{diag}(1.0000,\ 1.0000,\ 0.4375,\ 0.4375)$.

The following two examples, taken from [17], verify the conclusion, although not in the same number of steps.

4. $A$ in [17, Example 1]:

$$A = \begin{bmatrix} 1 & -0.2 & -0.1 & -0.2 & -0.1 \\ -0.4 & 1 & -0.2 & -0.1 & -0.1 \\ -0.9 & -0.2 & 1 & -0.1 & -0.1 \\ -0.3 & -0.7 & -0.3 & 1 & -0.1 \\ -1 & -0.3 & -0.2 & -0.4 & 1 \end{bmatrix}.$$

: "$A$ IS an $H$-matrix," $\max_{i=1(1)5} s_i^{(6)} = 0.9989 < 1$,
$D = \text{diag}(0.4178,\ 0.4802,\ 0.6560,\ 0.7648,\ 1)$.

5. $A$ in [17, Example 2]:

$$A = \begin{bmatrix} 1 & -0.8 & -0.1 \\ -0.5 & 1 & \mathrm{c} \\ -0.8 & -0.6 & 1 \end{bmatrix}.$$

For $c = -0.3951$:
: "$A$ IS an $H$-matrix," $\max_{i=1(1)3} s_i^{(8)} = 0.99999417061559 < 1$,
$D = \text{diag}(0.69344055479302,\ 0.74176649875408,\ 0.99985294182613)$.
For $c = -0.3952$:
: "$A$ is NOT an $H$-matrix," $\min_{i=1(1)3} s_i^{(9)} = 1.00001588177980 > 1$,
$D = \text{diag}(0.69343749916264,\ 0.74183369108397,\ 0.99983433483175)$.

Obviously, to the accuracy of four decimal places we consider $c$, it is clear that for all $|c| \leq 0.3951$ $A$ IS an $H$-matrix, while for $|c| \geq 0.3952$ $A$ is NOT an $H$-matrix.

The example below verifies the conclusion drawn by a Gauss–Seidel-type modification of Algorithm $\mathbb{H}$ of [16] given in an extended compact profile technique in [9]. In addition the case of inconclusiveness we had in [9] is now removed.

6. $A$ in the example of [9],

$$A = \begin{bmatrix} -1 & a_{12} & 0 & 0 & 0 \\ 0.5 & -1 & 0 & -0.6 & 0 \\ 0 & -0.1 & 1 & 0 & 0.5 \\ 0 & 0.5 & 0 & 1 & -0.5 \\ -0.2 & 0.1 & 0.3 & 0 & -1 \end{bmatrix}.$$

For $a_{12} = 1.146391$:
: "$A$ IS an $H$-matrix," $\max_{i=1(1)5} s_i^{(32)} = 0.99999993216569 < 1$,
$D = \text{diag}(1,\ 0.87230267174610,\ 0.27158312363400,\ 0.62050421587928,\ 0.36870533832715)$.
For $a_{12} = 1.146392$:

: "$A$ is NOT an $H$-matrix," $\min_{i=1(1)5} s_i^{(37)} = 1.00000002036218 > 1$, $D = \mathrm{diag}(1,\ 0.87230203336695,\ 0.27158269490209,\ 0.62050348290098,\ 0.36870499419081)$.

A similar note to the one in Example 6 is made. To the accuracy of six decimal places for $a_{12}$, for all $|a_{12}| \leq 1.146391$ $A$ IS an $H$-matrix, while for $|a_{12}| \geq 1.146392$ $A$ is NOT an $H$-matrix.

In the following two examples the given irreducible matrices, one of which is complex, are singular. Even in the first example this is indirectly spotted by our MATLAB function despite the fact that the output is "$A$ IS an $H$-matrix."

7. $A$ is the following matrix:

$$A = \begin{bmatrix} 2 & -1 & -0.5 \\ -1 & 2 & -1 \\ -0.5 & -1 & \frac{7}{6} \end{bmatrix}.$$

: "$A$ IS an $H$-matrix,"
$\min_{i=1(1)3} s_i^{(33)} = \max_{i=1(1)3} s_i^{(33)} = 1.0000$,
$D = \mathrm{diag}(0.6667,\ 0.8333,\ 1.0000)$.

8. $A$ is the following matrix:

$$A = \begin{bmatrix} \frac{1+\mathrm{i}\sqrt{3}}{4} & 2\sqrt{2}(1+\mathrm{i}) \\ \frac{\sqrt{2}(1-\mathrm{i})}{8} & 2(1-\mathrm{i}\sqrt{3}) \end{bmatrix}.$$

: "$\mathcal{M}(A)$ (to the MATLAB precision) IS SINGULAR,"
$\min_{i=1(1)2} s_i^{(3)} = \max_{i=1(1)2} s_i^{(3)} = 1$, $D = \mathrm{diag}(1.0000,\ 0.1250)$.

The last example is the one in Example 6 with $a_{12} = 1.146391$ except that $a_{33}$ was set to zero to check whether our MATLAB function can spot the presence of zero element(s) in the diagonal of the input matrix $A$.

9. $A_4$ in (2.12) except that $a_{33}$ is set to zero:

: "$A$ is NOT an $H$-matrix; It has at least one zero diagonal element,"
$\min_{i=1(1)4} s_i^{(0)} = 0$, $\max_{i=1(1)4} s_i^{(0)} = \infty$, $D = []$.

To conclude our work we make the following two points.

(i) Our algorithm (and MATLAB function), with a slight modification, works in case we want to find a good approximation to the spectral radius of an irreducible matrix with diagonal elements of the same sign and nonpositive (or nonnegative) off-diagonal elements. It can also find the spectral radius of the Jacobi matrix associated with a nonnegative (or nonpositive) irreducible matrix. In any of these cases a criterion like, e.g., $S^{(k)} - s^{(k)} < \eta$, where $\eta$ is the accuracy required for the spectral radius, should be set.

(ii) Our algorithm (and MATLAB function) works as is in case the matrix $A$ IS an $H$-matrix regardless of $A$ being irreducible or reducible. A possible extension of our present algorithm to also fully cover the reducible case is under investigation.

**Acknowledgments.** The authors would like to express their sincere thanks to the two unknown referees for their very constructive criticism, to the associate editor, Prof. Daniel Szyld, for the discussions the second author had with him on a first version of the paper as well as for various suggestions on later versions, to Prof. Volker Mehrmann for suggesting [22], and also to Prof. Michael Neumann for some useful comments. To all who contributed to the improvement of the quality of the presentation of this work the authors are most indebted.

## REFERENCES

[1] B.H. Ahn, *Solution of nonsymmetric linear complementarity problems by iterative methods*, J. Optim. Theory Appl., 33 (1981), pp. 175–185.

[2] A. Berman and R.J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[3] R. Bru and M. Neumann, *Nonnegative Jordan basis*, Linear Multilinear Algebra, 23 (1988), pp. 95–109.

[4] Lj. Cvetković and V. Kostić, *New criteria for identifying $H$-matrices*, J. Comput. Appl. Math., 180 (2005), pp. 265–278.

[5] D.K. Faddeev and V.N. Faddeeva, *Computational Methods of Linear Algebra*, W.H. Freeman, San Francisco, 1963.

[6] T.-B. Gan and X.-H. Huang, *Simple criteria for nonsingular $H$-matrices*, Linear Algebra Appl., 374 (2003), pp. 317–326.

[7] Y.-M. Gao and X.-H. Wang, *Criteria for generalized diagonally dominant matrices and $M$-matrices*, Linear Algebra Appl. 169 (1992), pp. 257–268.

[8] Y.-M. Gao and X.-H. Wang, *Criteria for generalized diagonally dominant matrices and $M$-matrices* II, Linear Algebra Appl., 248 (1996), pp. 339–353.

[9] A. Hadjidimos, *An extended compact profile iterative method criterion for sparse $H$-matrices*, Linear Algebra Appl., 389 (2004), pp. 329–345.

[10] M. Harada, M. Usui, and H. Niki, *An extension of the criteria for generalized diagonally dominant matrices*, Int. J. Comput. Math., 60 (1996), pp. 115–119.

[11] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[12] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.

[13] T.-Z. Huang, *A note on generalized diagonally dominant matrices*, Linear Algebra Appl., 225 (1995), pp. 237–242.

[14] D.R. Kincaid, J.R. Respess, D.M. Young, and R.G. Grimes, *ITPACK 2C: A Fortran package for solving large sparse linear systems by adaptive accelerated iterative methods*, ACM Trans. Math. Software, 8 (1982), pp. 302–322.

[15] T. Konho, H. Niki, H. Sawami, and Y.-M. Gao, *An iterative test for $H$-matrix*, J. Comput. Appl. Math., 115 (2000), pp. 349–355.

[16] B. Li, L. Li, M. Harada, H. Niki, and M.J. Tsatsomeros, *An iterative criterion for $H$-matrices*, Linear Algebra Appl., 271 (1998), pp. 179–190.

[17] L. Li, *On the iterative criterion for generalized diagonally dominant matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 17–24.

[18] L. Li, *Personal communication*, 2006.

[19] K. Ojiro, H. Niki, and M. Usui, *A new criterion for $H$-matrices*, J. Comput. Appl. Math., 150 (2003), pp. 293–302.

[20] A.M. Ostrowski, *Über die Determinanten mit Überwiegender Hauptdiagonale*, Comment. Math. Helv., 10 (1937), pp. 69–96.

[21] U.G. Rothblum, *Algebraic eigenspaces of nonnegative matrices*, Linear Algebra Appl., 12 (1975), pp. 281–292.

[22] H. Schneider, *The influence of the marked reduced graph of a nonnegative matrix on the Jordan form and related properties: A survey*, Linear Algebra Appl., 84 (1986), pp. 161–189.

[23] M.J. Tsatsomeros, *Personal communication*, 2006.

[24] R.S. Varga, *Matrix Iterative Analysis. Second Revised and Expanded Edition*, Springer-Verlag, Berlin, 2000.

[25] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.

[26] D.M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 177–183

# SOLVING REAL LINEAR SYSTEMS WITH THE COMPLEX SCHUR DECOMPOSITION*

CARLA D. MORAVITZ MARTIN[†] AND CHARLES F. VAN LOAN[‡]

**Abstract.** If the complex Schur decomposition is used to solve a real linear system, then the computed solution generally has a complex component because of roundoff error. We show that the real part of the computed solution that is obtained in this way solves a nearby *real* linear system. Thus, it is "numerically safe" to obtain real solutions to real linear systems via the complex Schur decomposition. This result is useful in certain Kronecker product situations where fast linear equation solving is made possible by reducing the involved matrices to their complex Schur form. This is critical because in these applications one cannot work with the real Schur form without greatly increasing the volume of work.

**Key words.** linear systems, Schur decomposition, back-substitution, Kronecker products

**AMS subject classifications.** 15A06, 65F05, 65G50

**DOI.** 10.1137/050631690

**1. Introduction.** The [illegible] states that if $A \in \mathbb{R}^{n\times n}$, then there exists a unitary $Q \in \mathbb{C}^{n\times n}$ so that $Q^HAQ = T$ is upper triangular. The eigenvalues that appear along the diagonal of $T$ can be arbitrarily ordered. See [3, p. 313].

This decomposition, coupled with back-substitution and matrix-vector multiplication, can be used to solve a real linear system $Ax = b$. Indeed, since $Q^Hb = (Q^HAQ)(Q^Hx) = T(Q^Hx)$ we have the following algorithm.

```
Algorithm SchurSolve
    Step 1. Compute the Schur decomposition $Q^HAQ = T$.
    Step 2. Form $c = Q^Hb$.
    Step 3. Solve $Ty = c$ by back-substitution.
    Step 4. Set $x = Qy$.
```

Ordinarily, it is preferred to work with the $LU$ factorization because it is much cheaper. However, there are settings involving Kronecker products when this is not the case. For example, the [illegible]

$$FX + XG^T = B, \qquad F \in \mathbb{R}^{m\times m},\ G \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{m\times n},$$

can be reshaped as $Ax = b$, where $A = I_n \otimes F + G \otimes I_m$, $x = \mathrm{vec}(X)$, and $b = \mathrm{vec}(B)$. (Here, $\mathrm{vec}(\cdot)$ makes a column vector out of a matrix by stacking its columns.) The $LU$ factorization of $A$ involves $O(m^3n^3)$ flops. But if we compute the Schur decompositions $Q_F^HFQ_F = R$ and $Q_G^HGQ_G = S$ and set $Q = Q_F \otimes Q_G$, then $Q^HAQ = I_n \otimes R + S \otimes I_m$ is the Schur decomposition of $A$ and `SchurSolve` requires $O(n^3 + m^3)$ flops if the Kronecker structure is exploited. See [3, p. 367].

---

*Received by the editors May 16, 2005; accepted for publication (in revised form) by B. T. Kagstrom July 17, 2006; published electronically December 21, 2006. This work was supported by NSF grant CCR-9901988.

http://www.siam.org/journals/simax/29-1/63169.html

[†] Center for Applied Mathematics, Cornell University, Ithaca, NY 14853-7510. Current address: Department of Mathematics and Statistics, James Madison University, MSC 1911, Harrisonburg, VA 22807 (carlam@math.jmu.edu).

[‡] Department of Computer Science, Cornell University, 4130 Upson Hall, Ithaca, NY 14853-7510 (cv@cs.cornell.edu).

A problem with `SchurSolve` is that complex arithmetic arises whenever $A$ has complex eigenvalues. This increases the volume of work. Moreover, the computed solution vector $x$ will inevitably have a complex component because of roundoff error. These problems can be avoided by working with the [illegible] Schur decomposition. In this factorization we find a real orthogonal $Q$ so that $Q^TAQ = T$ is upper quasi-triangular, i.e., block triangular with 1-by-1 and 2-by-2 diagonal blocks. Because $T$ is "almost" triangular, the `SchurSolve` philosophy essentially applies, except that a quasi-triangular system is solved in Step 3 of `SchurSolve`, rather than a (complex) triangular system.

Therefore, an algorithm to solve a linear system using the real Schur decomposition appears to involve a simple modification of `SchurSolve`. However, there are situations where the real Schur decomposition is much more expensive to compute than the (complex) Schur decomposition. Consider (again) the Sylvester equation problem. If we have computed the real Schur decompositions $Q_F^TFQ_F{=}R$ and $Q_G^TGQ_G = S$ and set $Q = Q_F \otimes Q_G$, then $Q^TAQ = I_n \otimes R + S \otimes I_m$ is [illegible] the real Schur decomposition of $A$. Attempting to compute the canonical form would destroy the Kronecker structure and would greatly increase the volume of work. Fortunately, there is a way of handling the subdiagonal blocks of $I_n \otimes R + S \otimes I_m$ using clever permutations so that the overall procedure remains $O(m^3 + n^3)$. See [3, p. 367].

However, in [4] we describe another Kronecker product situation where the permutation "device" does not work—specifically, the shifted Kronecker product system

$$\left(A^{(p)} \otimes \cdots \otimes A^{(1)} - \lambda I_N\right) x = b, \qquad \lambda \in \mathbb{R},\ b \in \mathbb{R}^N,\ N = n_1 \cdots n_p\ , \tag{1.1}$$

where $A^{(i)} \in \mathbb{R}^{n_i \times n_i}$ for $i = 1, \ldots, p$. After computing the real Schur decompositions of the $A^{(i)}$, a fast recursive procedure exists to solve for $x$ if the $A^{(i)}$ have real eigenvalues. However, if the $A^{(i)}$ have complex eigenvalues, the resulting $p$-fold Kronecker product of quasi-triangular matrices has a complicated and very problematic block structure below the diagonal, thus increasing the volume of work needed by the recursive procedure. This impasse brings us back to `SchurSolve` and the main contribution of this paper. In particular, we examine the properties of the real part of the computed solution $\hat{x}$.

The analysis to determine if a computed solution of a system solves a nearby system of the same form is illustrative of recent work in the general area of "structured" perturbations and error analysis. For example, in [5, 6] the conditioning of structured linear systems is examined where the structure includes symmetric, Toeplitz, circulant, and Hankel matrices. In addition, [1] analyzes the stability of algorithms for solutions of symmetric indefinite systems. In our paper, we show that the real part of the computed solution solves a nearby [illegible] system, a type of structured perturbation. We are not the first to examine complex algorithms for real problems. For example, [2] compares the condition of a complex eigenvalue of a real matrix under real and complex perturbations in order to analyze the accuracy of real algorithms versus complex algorithms. Our work is in this spirit, expanding what we know about structured perturbations for the case when "structure" means real data.

In section 2 we show that `SchurSolve` produces a complex solution that solves a nearby, but complex, linear system. This result is not new but is included for the sake of completeness. We then proceed to prove a perturbation theorem in section 3. It shows that when a real linear system is subjected to complex perturbations, then the real part of the solution to the perturbed system solves a nearby real linear system. This is followed by a brief summary in section 4.

Throughout this paper we use the 2-norm. The 2-norm condition of a matrix $M$ is denoted by $\kappa(M)$. The unit roundoff is designated by $\mathbf{u}$. We repeatedly use the fact that if $M \in \mathbb{C}^{m\times n}$, then both $\|\operatorname{Re}(M)\|$ and $\|\operatorname{Im}(M)\|$ are bounded by $\|M\|$.

**2. Backward error analysis.** We show that if $\hat{x}$ is the solution produced by `SchurSolve` when floating point arithmetic is used, then

$$(A+\Delta A)\,\hat{x} = b+\Delta b, \tag{2.1}$$

$$\|\Delta A\| \le \delta_A \|A\|, \tag{2.2}$$

$$\|\Delta b\| \le \delta_b \|b\|, \tag{2.3}$$

where the $\delta$'s are modest multiples of the unit roundoff $\mathbf{u}$. To present an uncluttered but sufficiently rigorous analysis, we adopt the convention that all the $\delta$'s below are $O(\mathbf{u})$ in magnitude. The floating point result of a matrix calculation is indicated by $\mathrm{fl}(\cdot)$. The floating point properties associated with the Schur decomposition, back-substitution, and other basic computations can be found in [3].

In Step 1 the computed Schur decomposition of $A \in \mathbb{R}^{n\times n}$ produces a "nearly" unitary $\hat{Q} \in \mathbb{C}^{n\times n}$. That is, there is an exactly unitary $Q \in \mathbb{C}^{n\times n}$ such that

$$Q = \hat{Q} + \Delta Q, \qquad \|\Delta Q\| \le \delta_1\ . \tag{2.4}$$

The computed Schur form $\hat{T}$ satisfies

$$\hat{T} = Q^H(A+H)Q, \qquad \|H\| \le \delta_2\|A\|\ , \tag{2.5}$$

where $H \in \mathbb{C}^{n\times n}$. Accounting for the roundoff error in Step 2, there exists $\Delta b \in \mathbb{C}^n$ such that

$$\hat{c} = \mathrm{fl}(\hat{Q}^H b) = Q^H(b+\Delta b), \qquad \|\Delta b\| \le \delta_b\|b\|, \tag{2.6}$$

while in Step 3 the computed solution to the triangular system satisfies

$$(\hat{T}+G)\hat{y} \;=\; \hat{c}, \qquad \|G\| \le \delta_3\|\hat{T}\| \le \delta_4\|A\|\ . \tag{2.7}$$

In the last step the computed solution $\hat{x}$ can be related to $\hat{y}$ as follows:

$$\hat{x} = \mathrm{fl}(\hat{Q}\hat{y}) \;=\; Q(\hat{y}+g), \qquad \|g\| \le \delta_5\|\hat{y}\| \le \delta_6\|\hat{x}\|\ . \tag{2.8}$$

Now let us combine these results. From (2.6) and (2.7) we have

$$(\hat{T}+G)\hat{y} \;=\; Q^H(b+\Delta b),$$

and so by (2.5) and (2.8)

$$b+\Delta b \;=\; Q\left(\hat{T}+G\right)Q^HQ\hat{y} \;=\; \left(A+H+QGQ^H\right)(\hat{x}-Qg)\,.$$

If $M = A+H+QGQ^H$, then

$$\begin{aligned} b+\Delta b = M(\hat{x}-Qg) \;&=\; \left(M - \frac{MQg\hat{x}^H}{\hat{x}^H\hat{x}}\right)\hat{x} \\ &= \left(A+H+QGQ^H - \frac{MQg\hat{x}^H}{\hat{x}^H\hat{x}}\right)\hat{x}, \end{aligned}$$

and so if we define

$$\Delta A = H + QGQ^H - \frac{MQg\hat{x}^H}{\hat{x}^H\hat{x}}, \tag{2.9}$$

then $(A + \Delta A)\,\hat{x} = b + \Delta b$, i.e., $\hat{x}$ solves a perturbed system. From (2.6) we know that $\Delta b$ satisfies (2.3). Thus, the verification of (2.1)–(2.3) is complete once we show that $\|\, \Delta A \,\|$ is sufficiently small. Toward that end we note that

$$\begin{aligned} \|\, M \,\| = \|\, A + H + QGQ^H \,\| &\le \|\, A \,\| + \|\, H \,\| + \|\, G \,\| \\ &\le (1 + \delta_2 + \delta_4)\|\, A \,\| = (1 + \delta_7)\|\, A \,\|. \end{aligned}$$

It follows from (2.5), (2.7), (2.8), and (2.9) that

$$\begin{aligned} \|\, \Delta A \,\| &\le \|\, H \,\| + \|\, G \,\| + \|\, M \,\| \frac{\|\, g \,\|}{\|\, \hat{x} \,\|} \\ &\le \delta_2\|\, A \,\| + \delta_4\|\, A \,\| + \delta_6(1 + \delta_7)\|\, A \,\|. \end{aligned}$$

The inequality (2.2) is established by setting $\delta_A = \delta_2 + \delta_4 + \delta_6(1 + \delta_7)$.

## 3. A perturbation theorem.

In this section we prove a result that will enable us to say something very favorable about the real part of the computed `SchurSolve` solution.

THEOREM 3.1. [illegible] $0 < \epsilon \le 1/6$, [illegible] $A \in \mathbb{R}^{n\times n}$, [illegible] $b \in \mathbb{R}^n$ [illegible] $\epsilon \cdot \kappa(A) \le 1/2$ [illegible]

$$(A + E)z = b + f, \tag{3.1}$$

[illegible]

$$\begin{array}{lll} E = E_1 + iE_2, & E_1,\, E_2 \in \mathbb{R}^{n\times n}, & \|\, E \,\| \le \epsilon\|\, A \,\|, \\ f = f_1 + if_2, & f_1,\, f_2 \in \mathbb{R}^n, & \|\, f \,\| \le \epsilon\|\, b \,\|, \\ z = z_1 + iz_2, & z_1,\, z_2 \in \mathbb{R}^n, & \end{array}$$

[illegible] $\tilde{E} \in \mathbb{R}^{n\times n}$ [illegible]

$$\left(A + \tilde{E}\right) z_1 = b + f_1 \tag{3.2}$$

[illegible]

$$\|\, \tilde{E} \,\| \le 4\epsilon\|\, A \,\|, \tag{3.3}$$

$$\|\, f_1 \,\| \le \epsilon\|\, b \,\| \,. \tag{3.4}$$

[illegible] Since

$$\|\, f_1 \,\| \le \|\, f_1 + if_2 \,\| = \|\, f \,\| \le \epsilon\|\, b \,\|,$$

the inequality (3.4) holds. Note that if $b = 0$ then $\|\, f \,\| = 0$ and so $\|\, f_1 \,\| = 0$. Expanding (3.1) we get

$$(A + E_1 + iE_2)(z_1 + iz_2) = b + f_1 + if_2$$

from which follows

$$(A+E_1)z_1 - E_2 z_2 = b + f_1, \tag{3.5}$$

$$(A+E_1)z_2 + E_2 z_1 = f_2. \tag{3.6}$$

If $b = 0$ and $z_1 = 0$, then any such $\tilde{E}$ such that $\| \tilde{E} \| \leq 4\epsilon \| A \|$ completes the proof. If $z_1 \neq 0$, (3.5) can be rewritten as

$$\left( A + E_1 - \frac{E_2 z_2 z_1^T}{z_1^T z_1} \right) z_1 = b + f_1,$$

and so (3.2) holds with

$$\tilde{E} = E_1 - \frac{E_2 z_2 z_1^T}{z_1^T z_1}. \tag{3.7}$$

Now to establish (3.3), we start by taking norms in (3.7):

$$\| \tilde{E} \| \leq \| E_1 \| + \| E_2 \| \frac{\| z_2 \|}{\| z_1 \|} \leq \epsilon \| A \| \left( 1 + \frac{\| z_2 \|}{\| z_1 \|} \right) . \tag{3.8}$$

Looking at (3.3), we must confirm that $\| z_2 \|$ is not too much bigger than $\| z_1 \|$. From (3.6) we have

$$z_2 = (A+E_1)^{-1} (f_2 - E_2 z_1) = (I + A^{-1}E_1)^{-1} A^{-1} (f_2 - E_2 z_1),$$

and so

$$\| z_2 \| \leq \| (I + A^{-1}E_1)^{-1} \| \| A^{-1} \| (\| f_2 \| + \| E_2 \| \| z_1 \|) .$$

The assumption $\epsilon \cdot \kappa(A) < 1/2$ implies

$$\| (I + A^{-1}E_1)^{-1} \| \leq \frac{1}{1 - \| A^{-1}E_1 \|} \leq \frac{1}{1 - \epsilon \cdot \| A \| \| A^{-1} \|} \leq 2,$$

and thus

$$\| z_2 \| \leq 2\epsilon \| A^{-1} \| (\| b \| + \| A \| \| z_1 \|) . \tag{3.9}$$

By rearranging (3.5) we see that $b = (A+E_1)z_1 - E_2 z_2 - f_1$ and therefore

$$\begin{aligned} \| b \| &\leq (\| A \| + \| E_1 \|) \| z_1 \| + \| E_2 \| \| z_2 \| + \| f_1 \| \\ &\leq (1+\epsilon) \| A \| \| z_1 \| + \epsilon \| A \| \| z_2 \| + \epsilon \| b \| \\ &\leq \frac{1+\epsilon}{1-\epsilon} \| A \| \| z_1 \| + \frac{\epsilon}{1-\epsilon} \| A \| \| z_2 \|. \end{aligned}$$

By substituting this inequality into (3.10) and using the assumption that $\epsilon \leq 1/6$ we get

$$\begin{aligned}
\| z_2 \| &\leq 2\epsilon \| A^{-1} \| \left( \frac{1+\epsilon}{1-\epsilon} \| A \| \| z_1 \| + \frac{\epsilon}{1-\epsilon} \| A \| \| z_2 \| \; + \; \| A \| \| z_1 \| \right) \\
&= 2\epsilon\kappa(A) \left( \frac{2}{1-\epsilon} \| z_1 \| \; + \; \frac{\epsilon}{1-\epsilon} \| z_2 \| \right) \\
&\leq \left( \frac{2}{1-\epsilon} \| z_1 \| \; + \; \frac{\epsilon}{1-\epsilon} \| z_2 \| \right) \\
&\leq \frac{2}{1-2\epsilon} \| z_1 \| \; \leq \; 3 \| z_1 \|.
\end{aligned}$$

The inequality (3.3) follows from this and (3.9).

The proof will be complete after we address whether $z_1$ can be zero. By way of contradiction, assume $z_1 = 0$. Then (3.5) and (3.6) become $-E_2 z_2 = b + f_1$ and $(A + E_1) z_2 = f_2$, respectively. So

$$\begin{aligned}
b &= -f_1 - E_2 z_2 \\
&= -f_1 - E_2 (A + E_1)^{-1} f_2.
\end{aligned}$$

This implies that

$$\begin{aligned}
\| b \| &\leq \| f_1 \| + \| E_2 \| \| (A + E_1)^{-1} \| \| f_2 \| \\
&\leq \epsilon \| b \| + 2\epsilon \| A \| \| A^{-1} \| (\epsilon \| b \|) \\
&\leq 2\epsilon \| b \|,
\end{aligned}$$

and therefore $1 \leq 2\epsilon$, which is a contradiction. Since $z_1$ is nonzero, the proof is now complete. □

In section 2 we showed that SchurSolve produces a computed solution $\hat{x}$ that exactly solves a ( ) linear system that is “within roundoff” of the original. Thus,

$$\frac{\| \hat{x} - x \|}{\| x \|} \; \approx \; \mathbf{u}\kappa(A)\,.$$

Since $\| \mathrm{Re}(\hat{x}) - x \| \leq \| \hat{x} - x \|$ it follows that

$$\frac{\| \mathrm{Re}(\hat{x}) - x \|}{\| x \|} \; \approx \; \mathbf{u}\kappa(A),$$

which is what we would expect from a stable linear equation solving process. But we can say more in light of Theorem 3.1. The following corollary to Theorem 3.1 shows that using the complex Schur decomposition to solve a real problem results in a computed solution whose real part solves a nearby real system.

COROLLARY 3.2. $A \in \mathbb{R}^{n \times n}$ $b \in \mathbb{R}^n$ $\hat{x}$ $Ax = b$ SchurSolve $\epsilon = \max(\delta_A, \delta_b) \leq 1/6$ $\Delta A \in \mathbb{R}^{n \times n}$ $\delta b \in \mathbb{R}^n$

$$(A + \Delta A)\mathrm{Re}(\hat{x}) = b + \delta b, \tag{3.10}$$

[illegible]

$$\| \Delta A \| \leq \delta_A \| A \|, \tag{3.11}$$

$$\| \delta B \| \leq \delta_b \| b \|. \tag{3.12}$$

**4. Summary.** We have illustrated certain situations where the complex Schur decomposition is preferred to using the real Schur decomposition when solving a real system. Thus, we show that it is numerically safe to obtain solutions to the real system by introducing complex arithmetic. In particular, Theorem 3.1 and Corollary 3.2 show that the real part of the computed solution obtained using `SchurSolve` solves a nearby [illegible] linear system.

## REFERENCES

[1] J. R. Bunch, J. W. Demmel, and C. F. Van Loan, *The strong stability of algorithms for solving symmetric linear systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 494–499.

[2] R. Byers and D. Kressner, *On the condition of a complex eigenvalue under real perturbations*, BIT, 44 (2004), pp. 209–214.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[4] C. D. Moravitz Martin and C. F. Van Loan, *Shifted Kronecker product systems*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 184–198.

[5] S. M. Rump, *Structured perturbations part* I: *Normwise distances*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 1–30.

[6] S. M. Rump, *Structured perturbations part* II: *Componentwise distances*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 31–56.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 184–198

# SHIFTED KRONECKER PRODUCT SYSTEMS*

CARLA D. MORAVITZ MARTIN[†] AND CHARLES F. VAN LOAN[‡]

**Abstract.** A fast method for solving a linear system of the form $(A^{(p)} \otimes \cdots \otimes A^{(1)} - \lambda I)x = b$ is given where each $A^{(i)}$ is an $n_i$-by-$n_i$ matrix. The first step is to convert the problem to triangular form $(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I)y = c$ by computing the (complex) Schur decompositions of the $A^{(i)}$. This is followed by a recursive back-substitution process that fully exploits the Kronecker structure and requires just $O(N(n_1 + \cdots + n_p))$ flops where $N = n_1 \cdots n_p$. A similar method is employed when the real Schur decomposition is used to convert each $A^{(i)}$ to quasi-triangular form. The numerical properties of these new methods are the same as if we explicitly formed $(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I)$ and used conventional back-substitution to solve for $y$.

**Key words.** linear systems, Schur decomposition, back-substitution, Kronecker products

**AMS subject classifications.** 15A06, 65F05, 65G50

**DOI.** 10.1137/050631707

## 1. Introduction.

Matrix problems with replicated block structure abound in signal and image processing, semidefinite programming, control theory, and many other application areas. In these venues fast algorithms have emerged that exploit the rich algebra of the Kronecker product. Perhaps the best example of this is the fast Fourier transform which can be described using the "language" of sparse matrix factorizations and the Kronecker product. This operation is surfacing more and more as cheap memory prompts the assembly of huge, multidimensional datasets. When techniques for problems of low dimension are generalized or "tensored" together to address a high-dimensional, multilinear problem, then one typically finds a computational challenge that involves the Kronecker product.

It is in the spirit of bringing the fruits of numerical linear algebra to the realm of numerical multilinear algebra that we present the current paper. Our goal is to present a methodology for solving a shifted linear system when the matrix of coefficients is a Kronecker product. Specifically, the question we address is how to solve a shifted Kronecker product system of the form

$$\left(A^{(p)} \otimes \cdots \otimes A^{(1)} - \lambda I_N\right) x = b, \qquad b \in \mathbb{R}^N, \tag{1.1}$$

where $A^{(i)} \in \mathbb{R}^{n_i \times n_i}$, $i = 1{:}p$, are given and $N = n_1 \cdots n_p$. A reshaped special case of this problem is the discrete-time Sylvester equation $A^{(1)} X {A^{(2)}}^T - X = B$. As with many matrix equations of this variety, the first step is to convert $A^{(1)}$ and $A^{(2)}$ to triangular form via the Schur decompoistion. The resulting system can then be solved via a back-substitution process. Jonsson and Kågström [2] have developed block

---

[*] Received by the editors May 16, 2005; accepted for publication (in revised form) by B. T. Kågström July 17, 2006; published electronically December 21, 2006. This work was supported by NSF grant CCR-9901988.

http://www.siam.org/journals/simax/29-1/63170.html

[†] Center for Applied Mathematics, Cornell University, Ithaca, NY 14853-7510. Current address: Department of Mathematics and Statistics, James Madison University, MSC 1911, Harrisonburg, VA 22807 (carlam@math.jmu.edu).

[‡] Department of Computer Science, Cornell University, 4130 Upson Hall, Ithaca, NY 14853-7510 (cv@cs.cornell.edu).

recursive methods for these kinds of problems and they are very effective in high-performance computing environments. The method we present is also recursive and can be regarded as a generalization of their technique. However, we do not generate the subproblems by splitting at the block level.

There are other well-known settings where linear equation solving via the Schur, real Schur, or Hessenberg decompositions is preferred over Gaussian elimination and the $LU$ factorization. For example, suppose $A \in \mathbb{R}^{N\times N}$, $b \in \mathbb{R}^N$, $d \in \mathbb{R}^N$, and that we want to explore the behavior of the function

$$f(\lambda) = d^T(A - \lambda I_N)^{-1}b,$$

where $\lambda$ is a scalar. Note that for each $\lambda$ we must solve a system of linear equations

$$(A - \lambda I_N)x = b. \tag{1.2}$$

If one proceeds to use Gaussian elimination, then each $f$-evaluation requires $\mathrm{O}(N^3)$ flops because the underlying $LU$ factorization must be recomputed from scratch for each $\lambda$.

If many $f$-evaluations are required, then a better approach is to rely on a similarity transformation such as the Schur or Hessenberg decomposition:

$$Q^H A Q = T. \tag{1.3}$$

Here $Q \in \mathbb{C}^{N\times N}$ is unitary, and $T \in \mathbb{C}^{N\times N}$ is upper triangular, quasi-triangular, or Hessenberg, depending on whether $A$ has complex eigenvalues and depending on whether the real or complex Schur (or Hessenberg) decomposition is used. Once this $O(N^3)$ "investment" is performed, then

$$f(\lambda) = \tilde{d}^T(T - \lambda I_N)^{-1}\tilde{b}, \qquad \tilde{b} = Q^T b,\ \tilde{d} = Q^T d,$$

can be evaluated in just $O(N^2)$ flops. In practice, one typically invokes the Hessenberg decomposition because it is cheaper, or the real Schur decomposition because it permits the handling of the complex eigenvalue case with real arithmetic.

Applying these ideas to (1.1) we first compute the Schur decompositions

$$Q^{(i)H} A^{(i)} Q^{(i)} \;=\; T^{(i)}, \qquad i = 1{:}p, \tag{1.4}$$

a calculation that requires $O(n_1^3 + \cdots + n_p^3)$ flops. If

$$Q = Q^{(p)} \otimes \cdots \otimes Q^{(1)},$$

then $Q$ is unitary and

$$Q^H\left(A^{(p)} \otimes \cdots \otimes A^{(1)}\right)Q \;=\; T^{(p)} \otimes \cdots \otimes T^{(1)}.$$

Thus, (1.1) transforms to

$$\left(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I_N\right) y = c, \tag{1.5}$$

where $y \in \mathbb{R}^N$ and $c \in \mathbb{R}^N$ are defined by

$$x \;=\; \left(Q^{(p)} \otimes \cdots \otimes Q^{(1)}\right) y \tag{1.6}$$

and

$$c = \left(Q^{(p)} \otimes \cdots \otimes Q^{(1)}\right)^H b. \tag{1.7}$$

If the Kronecker structure is exploited, then the computations for $x$ and $c$ require $O(N(n_1 + \cdots + n_p))$ flops. If the complex Schur decomposition is used, then the resulting system (1.5) is triangular, and we show that it can also be solved in $O(N(n_1 + \cdots + n_p))$ flops. If the real Schur decomposition is used, then the Kronecker product in (1.5) has a complicated structure. In this case, we invoke the complex Schur decomposition to deal with the 2-by-2 bumps in each of the $T^{(i)}$. Regardless, the system (1.5) is an example where introducing complex arithmetic to solve a real problem is more advantageous. Our main contribution is to show that we can solve (1.5) "just as fast" where the $T^{(i)}$ are either upper triangular or upper quasi-triangular. In both cases, complex operations are used to solve the problem.

Thus, with our new method in place, the overall solution process (1.4)–(1.7) requires just $O(N(n_1 + \cdots + n_p))$ flops to execute. To put this in perspective, $O(N^2)$ flops are typically needed for $N$-by-$N$ triangular system solving and $O(N^3)$ flops for the preliminary factorization. Note that we are assuming that the Schur decompositions in (1.4) are insignificant. Exceptions occur, for example, when $p = 2$ and $n_1 \gg n_2$.

We stress that it is the presence of the shift $\lambda$ in (1.5) that creates the problem. If $\lambda = 0$, then we have an easy factorization of the matrix of coefficients. Indeed, if the $T^{(i)}$ are upper triangular,

$$\left(T^{(p)} \otimes \cdots \otimes T^{(1)}\right) = \prod_{i=1}^{p} \left(I_{\rho_i} \otimes T^{(i)} \otimes I_{\mu_i}\right),$$

where $\rho_i = n_{i+1} \cdots n_p$ and $\mu_i = n_1 \cdots n_{i-1}$ for $i = 1{:}p$. A sequence of triangular system solves can then be used to obtain $y$:

(1.8)
```
$y \leftarrow c$
for $i = 1{:}p$
    $y \leftarrow \left(I_{\rho_i} \otimes T^{(i)} \otimes I_{\mu_i}\right)^{-1} y$
end
```

This implementation of back-substitution requires $N(n_1 + \cdots + n_p)$ flops.

Unfortunately, if $\lambda \neq 0$ then we are stranded without a "Kronecker-friendly" factorization for $\left(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I_N\right)$. However, we can implement a recursive back-substitution procedure involving the Schur decomposition so that (1.5) can be solved as fast as (1.8).

Our presentation is structured as follows. In section 2 we review relevant properties of the Kronecker product. To motivate our general procedure for both the triangular and quasi-triangular case, we consider the $p = 2$ case in section 3. In section 4 we present the algorithm for general $p$ using both the complex Schur decomposition and the real Schur decomposition. Numerical behavior and various performance and implementation issues are discussed at the end in section 5. Finally, the error analysis is presented in the appendix.

## 2. Some properties of the Kronecker product.

We review a few essential facts about the Kronecker product. Details and proofs can be found in [4].

Matrix computations that involve the Kronecker product require an understanding of the `vec` and `reshape` operators. If $Z \in \mathbb{R}^{m\times n}$, then the `vec` operator is defined by

$$\texttt{vec}(Z) = \begin{bmatrix} Z(:,1) \\ \vdots \\ Z(:,n) \end{bmatrix} \in \mathbb{R}^{mn}.$$

In other words, $\texttt{vec}(Z)$ is a vector obtained by stacking the columns of $Z$.

The `reshape` operator is a more general way of rearranging the entries in a matrix. (It is also a built-in MATLAB function.) If $z \in \mathbb{R}^{mn}$ then $Z = \texttt{reshape}(z, m, n)$ is the $m$-by-$n$ matrix defined by

$$Z(:,j) = z(1 + (j-1)m{:}jm), \qquad j = 1{:}n.$$

For example, if $m = 3$ and $n = 5$, then

$$\texttt{reshape}(z,3,5) = \begin{bmatrix} z_1 & z_4 & z_7 & z_{10} & z_{13} \\ z_2 & z_5 & z_8 & z_{11} & z_{14} \\ z_3 & z_6 & z_9 & z_{12} & z_{15} \end{bmatrix} = Z.$$

Thus, $\texttt{reshape}(z, m, n)$ makes a matrix out of $z$ by using its components to "fill up" an $m$-by-$n$ array in column-major order. We also use `reshape` to build new matrices from the components of a given matrix. If $Z \in \mathbb{R}^{m_1\times n_1}$ and $m_2 n_2 = m_1 n_1$, then $\texttt{reshape}(Z, m_2, n_2)$ is the $m_2$-by-$n_2$ matrix $\texttt{reshape}(\texttt{vec}(Z), m_2, n_2)$.

If $F$, $G$, $H$, and $K$ are matrices and the multiplications $FH$ and $GK$ are defined, then $(F \otimes G)(H \otimes K) = FH \otimes GK$. Moreover, $(F \otimes G)^{-1} = (F^{-1} \otimes G^{-1})$ and $(F \otimes G)^T = F^T \otimes G^T$, assuming in the former case that $F$ and $G$ are nonsingular.

In general, if $F \in \mathbb{R}^{m\times m}$ and $G \in \mathbb{R}^{n\times n}$ then $F \otimes G \neq G \otimes F$. However, if we define the permutation matrix $\Pi_{n,nm} \in \mathbb{R}^{mn\times mn}$ by

$$\Pi^T_{n,nm} x = \begin{bmatrix} x(1{:}n{:}nm) \\ x(2{:}n{:}nm) \\ \vdots \\ x(n-1{:}n{:}nm) \end{bmatrix},$$

then it can be shown that

$$\Pi^T_{n,nm}(F \otimes G)\Pi_{n,nm} = G \otimes F.$$

The matrix $\Pi_{n,nm}$ is called the [illegible] and its action on a vector is very neatly described in terms of the `reshape` operation:

$$y = \Pi^T_{n,mn} x \quad \Leftrightarrow \quad \texttt{reshape}(y, m, n) = \texttt{reshape}(x, n, m)^T, \tag{2.1}$$

$$y = \Pi_{n,mn} x \quad \Leftrightarrow \quad \texttt{reshape}(y, n, m) = \texttt{reshape}(x, m, n)^T. \tag{2.2}$$

Note that $y = \Pi_{2,52} x$ is the perfect shuffle of the "card deck" $x \in \mathbb{R}^{52}$. We mention that if $x$ (and $y$) are complex, then (2.1) and (2.2) apply exactly as they are specified; the transpose is [illegible] replaced by a conjugate transpose.

The `vec` operator enables us to identify certain matrix-vector products as matrix-matrix products. In particular, if $F \in \mathbb{R}^{m\times m}$, $G \in \mathbb{R}^{n\times n}$, and $X \in \mathbb{R}^{n\times m}$, then it can be shown that

$$Y = GXF^T \qquad \Leftrightarrow \qquad \texttt{vec}(Y) = (F \otimes G)\,\texttt{vec}(X). \tag{2.3}$$

For matrix-vector products of the form

$$y = (F_p \otimes \cdots \otimes F_1)\, x, \qquad F_i \in \mathbb{R}^{n_i\times n_i}, \tag{2.4}$$

it is convenient to make use of the factorization

$$F_p \otimes \cdots \otimes F_1 \;=\; M_p \cdots M_1, \tag{2.5}$$

where

$$M_i = \Pi^T_{n_i,N}\left(I_{N/n_i} \otimes F_i\right) \tag{2.6}$$

and $N = n_1 \cdots n_p$. This result can be found in [4, p. 153] where it is exploited in connection with high-dimensional fast Fourier transforms. In practice, here is how one typically computes the vector $y$ in (2.4):

(2.7)

```
$Z \leftarrow x$
for $i = 1{:}p$
    $Z \leftarrow (F_i \cdot \texttt{reshape}(Z, n_i, N/n_i))^T$
end
$y \leftarrow \texttt{reshape}(Z, N, 1)$
```

The $i$th pass through the loop requires $(2n_i^2)(N/n_i) = 2Nn_i$ flops so the overall computation involves $2N(n_1 + \cdots n_p)$ flops.

We mention that a similar process can be used to solve

$$(F_p \otimes \cdots \otimes F_1)\, x = d.$$

From (2.5) and (2.6) it follows that

$$(F_p \otimes \cdots \otimes F_1)^{-1} \;=\; F_p^{-1} \otimes \cdots \otimes F_1^{-1} \;=\; M_1^{-1} \cdots M_p^{-1},$$

where

$$M_i^{-1} = \left(I_{N/n_i} \otimes F_i^{-1}\right)\Pi_{n_i,N},$$

and so we obtain

(2.8)

```
$B \leftarrow d$
for $i = p{:} - 1{:}1$
    $B \leftarrow F_i^{-1}\, \texttt{reshape}(B, N/n_i, n_i)^T$
end
$x \leftarrow \texttt{reshape}(B, N, 1)$
```

If the $F_i$ are triangular, then the $i$th pass through the loop requires $n_i^2(N/n_i) = Nn_i$ flops.

**3. The $p = 2$ case.** To motivate the proposed new method for (1.5) for the triangular and quasi-triangular case, we first consider the special case when $p = 2$. That is, for $F \in \mathbb{R}^{m\times m}$ and $G \in \mathbb{R}^{n\times n}$,

$$(F \otimes G - \lambda I)y = c. \tag{3.1}$$

Using (2.3), one can rewrite (3.1) as the real discrete-time Sylvester matrix equation

$$GYF^T - \lambda Y = C,$$

where $Y = \texttt{reshape}(y, n, m)$ and $C = \texttt{reshape}(c, n, m)$. As we mentioned earlier, a block procedure for solving the real discrete-time Sylvester matrix equation is described in [2]. In this section, we describe the details of solving (3.1) in a way that facilitates the presentation of the general $p > 2$ algorithm. We begin with a small particular problem, $(F \otimes G - \lambda I_{3n})\, y = c$, where

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ 0 & f_{22} & f_{23} \\ 0 & 0 & f_{33} \end{bmatrix}$$

and $G \in \mathbb{R}^{n\times n}$ is upper triangular. The shifted Kronecker system has the form

$$\begin{bmatrix} f_{11}G - \lambda I_n & f_{12}G & f_{13}G \\ 0 & f_{22}G - \lambda I_n & f_{23}G \\ 0 & 0 & f_{33}G - \lambda I_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix},$$

where $y_i \in \mathbb{R}^n$ and $c_i \in \mathbb{R}^n$ for $i = 1{:}3$. Assume that the system is nonsingular. The first step is to solve the $n$-by-$n$ triangular system

$$(f_{33}G - \lambda I_n)\, y_3 = c_3$$

for $y_3$. By substituting $y_3$ into the first two equations we obtain

$$\begin{bmatrix} f_{11}G - \lambda I_n & f_{12}G \\ 0 & f_{22}G - \lambda I_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \tilde{c}_1 \\ \tilde{c}_2 \end{bmatrix}, \tag{3.2}$$

where $\tilde{c}_i = c_i - f_{i3}Gy_3$, $i = 1{:}2$. The vectors $y_3$ and $Gy_3$ require $O(n^2)$ flops. This process is then repeated to render $y_2$, and $y_1$ in turn.

Note from the example that if $G$ is quasi-triangular (or even Hessenberg), then the systems involving the $(f_{ii}G - \lambda I_n)$ still just require $O(n^2)$ flops to solve. However, if $F$ is upper quasi-triangular, then there is a more serious complication. To illustrate let us examine the system $(F \otimes G - \lambda I_{4n})\, y = c$, where

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ 0 & f_{22} & f_{23} & f_{24} \\ 0 & f_{32} & f_{33} & f_{34} \\ 0 & 0 & 0 & f_{44} \end{bmatrix}$$

and $G \in \mathbb{R}^{n\times n}$ is upper quasi-triangular. In this case the shifted Kronecker system has the form

$$\begin{bmatrix} f_{11}G - \lambda I_n & f_{12}G & f_{13}G & f_{14}G \\ 0 & f_{22}G - \lambda I_n & f_{23}G & f_{24}G \\ 0 & f_{32}G & f_{33}G - \lambda I_n & f_{34}G \\ 0 & 0 & 0 & f_{44}G - \lambda I_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix},$$

where $y_i \in \mathbb{R}^n$ and $c_i \in \mathbb{R}^n$ for $i = 1{:}4$. Assume that the system is nonsingular. The first step is to solve the $n$-by-$n$ quasi-triangular system

$$(f_{44}G - \lambda I_n)\, y_4 = c_4$$

for $y_4$. Substituting this into the above system reduces it to

$$\begin{bmatrix} f_{11}G - \lambda I_n & f_{12}G & f_{13}G \\ 0 & f_{22}G - \lambda I_n & f_{23}G \\ 0 & f_{32}G & f_{33}G - \lambda I_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \tilde{c}_1 \\ \tilde{c}_2 \\ \tilde{c}_3 \end{bmatrix}, \tag{3.3}$$

where $\tilde{c}_i = c_i - f_{i4}Gy_4$. The vectors $y_4$ and $Gy_4$ require $O(n^2)$ flops. Next we solve the block 2-by-2 system

$$\begin{bmatrix} f_{22}G - \lambda I_n & f_{23}G \\ f_{32}G & f_{33}G - \lambda I_n \end{bmatrix} \begin{bmatrix} y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \tilde{c}_2 \\ \tilde{c}_3 \end{bmatrix} \tag{3.4}$$

for $y_2$ and $y_3$. We are then left with a single system for $y_1$:

$$(f_{11}G - \lambda I_n)y_1 = \tilde{c}_1 - f_{12}Gy_2 - f_{13}Gy_3.$$

From this example the general plan is clear. At each stage we solve either an $n$-by-$n$ system for a single $y_i$ or a $2n$-by-$2n$ block system for a pair of $y_i$'s. The results are then substituted into the remaining equations.

Now let us consider how to solve a system of the form (3.4). For concreteness, suppose

$$G = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}.$$

It is easy to verify that the 2-by-2 block matrix of coefficients in (3.4) has the form

$$\left[\begin{array}{ccccc|ccccc} \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times & 0 & 0 & 0 & \times & \times \\ \hline \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times & \times & 0 & 0 & 0 & \times & \times \end{array}\right] = S. \tag{3.5}$$

Since $S = F(2{:}3, 2{:}3) \otimes G - \lambda I_{10}$, we can reverse the order of the Kronecker factors via a permutation as discussed in section 2:

$$\Pi_{5,10}^T S \Pi_{5,10} = G \otimes F(2{:}3, 2{:}3) - \lambda I_{10} = \left[\begin{array}{cccc|cc|cccc} \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & 0 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times & \times & \times & \times \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \times & \times & \times & \times \end{array}\right].$$

Extrapolating from this example it is clear that a block system like (3.4) can be solved in $O(n^2)$ flops by permuting it into a block triangular system with diagonal blocks that are either 2-by-2 or 4-by-4.

We are now in a position to formulate a complete algorithm for the problem $(F \otimes G - \lambda I_{mn})y = c$ when $F \in \mathbb{R}^{m\times m}$ and $G \in \mathbb{R}^{n\times n}$ are upper quasi-triangular. If

$$F = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1r} \\ 0 & F_{22} & \cdots & F_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{rr} \end{bmatrix}$$

with 1-by-1 and 2-by-2 diagonal blocks, then $(F \otimes G - \lambda I_{mn})y = c$ has the form

$$\begin{bmatrix} F_{11} \otimes G - \lambda I_\ell & F_{12} \otimes G & \cdots & F_{1r} \otimes G \\ 0 & F_{22} \otimes G - \lambda I_\ell & \cdots & F_{2r} \otimes G \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & F_{rr} \otimes G - \lambda I_\ell \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_r \end{bmatrix},$$

where $\ell = n$ if $F_{ii}$ is 1-by-1 and $\ell = 2n$ if $F_{ii}$ is 2-by-2 for $i = 1{:}r$. The overall back-substitution process then looks like this:

```
(3.6)
for $k = r{:}-1{:}1$
    if $F_{kk}$ is 1-by-1
        Solve $(F_{kk}G - \lambda I_n)y_k = c_k$ for $y_k \in \mathbb{R}^n$
        $z \leftarrow Gy_k$
        $c_i \leftarrow c_i - F_{ik}z,\ \ i = 1{:}k-1$
    else
        Solve $(F_{kk} \otimes G - \lambda I_{2n})y_k = c_k$ for $y_k \in \mathbb{R}^{2n}$
        $z \leftarrow (I_2 \otimes G)y_k$
        $c_i \leftarrow c_i - (F_{ik} \otimes I_n)z,\ \ i = 1{:}k-1$
    end
end
```

Thus, on each pass through the loop we solve an $n$-by-$n$ quasi-triangular system or a $2n$-by-$2n$ block triangular system obtained via permutation. The exact flop count depends upon the number of 2-by-2 blocks along the diagonals of $F$ and $G$, i.e., the number of complex conjugate eigenvalue pairs that these matrices have. But regardless, the volume of computation is $O(mn(m+n))$.

**4. The general algorithm.** Observe that algorithm (3.6) could be a solution framework for the general $(T^{(p)} \otimes \cdots \otimes T^{(1)})y = c$ problem if we set

$$\begin{aligned} F &= T^{(p)}, \\ G &= T^{(p-1)} \otimes \cdots \otimes T^{(1)}, \\ m &= n_p, \\ n &= n_1 \cdots n_{p-1}. \end{aligned}$$

The "solve" steps in (3.6) become recursive calls. If the $T^{(i)}$ are all upper triangular, then $F_{kk}$ is 1-by-1 and algorithm (3.6) can be easily extended for general $p$. However, if the $T^{(i)}$ are quasi-triangular, then $F_{kk}$ is 2-by-2 and the system $(F_{kk} \otimes G - \lambda I_{2n})y_k = c_k$ has the form

$$\left(F_{kk} \otimes T^{(p-1)} \otimes \cdots \otimes T^{(1)} - \lambda I_m\right) y_k = c_k.$$

If we use the methods of the previous section, we can permute this system to obtain

$$\left(T^{(p-1)} \otimes \cdots \otimes T^{(1)} \otimes F_{kk} - \lambda I_m\right) \tilde{y}_k = \tilde{c}_k. \tag{4.1}$$

However, the permute-to-block-triangular-form approach that we illustrated in section 3 is much less appealing when we consider the general $p$ case. If $G$ is itself a Kronecker product, e.g., $T^{(p-1)} \otimes \cdots \otimes T^{(1)}$, then its structure is adversely scrambled when we permute $S$ in (3.5).

For this reason, if we are confronted with a system of the form

$$(\alpha \otimes G - \lambda I)\, y = c, \tag{4.2}$$

where

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}$$

has complex eigenvalues, then we compute the [illegible] 2-by-2 Schur decomposition

$$Q^H \alpha Q = \begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix}.$$

Equation (4.2) transforms to

$$\left(\begin{bmatrix} s_{11} & s_{12} \\ 0 & s_{22} \end{bmatrix} \otimes G - \lambda I\right) z = d,$$

where $z = (Q^H \otimes I)y$ and $d = (Q^H \otimes I)c$. This can be solved recursively when $G$ is a Kronecker product. The (real) solution to the original system is then prescribed by $y = (Q \otimes I)z$.

Therefore, if the $T^{(i)}$ are upper quasi-triangular, extending algorithm (3.6) for general $p$ involves creating an input parameter, $\alpha$, that can be either a 1-by-1 or a 2-by-2 matrix. If $\alpha$ is 2-by-2 we compute its complex Schur decomposition and solve

$$\left(\begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \otimes T^{(p-1)} \otimes \cdots \otimes T^{(1)} - \lambda I\right) y = c$$

recursively.

Of course, the hassle associated with the 2-by-2 bumps can be avoided altogether if the complex Schur decompositions $Q^{(i)H}AQ^{(i)} = T^{(i)}$ are computed right at the start. However, the proposed strategy is preferred because it restricts complex arithmetic to diagonal block subproblems. In pseudo-MATLAB our algorithm, `KPShiftSolve`, is given in Figure 1.

To assess the volume of the computation, let $\nu_p$ be the number of flops required by a call to `KPShiftSolve` when the matrix of coefficients involves a $p$-fold Kronecker product. Ignoring low-order terms,

$$\nu_p = \begin{cases} 1.5n_1^2 & \text{if } p = 1, \\ \\ n_p\nu_{p-1} + n_1 \cdots n_p(n_1 + \cdots + n_p) & \text{if } p > 1. \end{cases} \tag{4.3}$$

Of course, the exact flop count depends on the number of complex eigenvalues of the $T^{(i)}$. We mentioned that an alternative, but more costly, algorithm involves computing the complex Schur decompositions of the $T^{(i)}$ from the start. If $p = 1$, then solving $(T_1 - \lambda I)y = c$ in complex arithmetic requires $6.5n_1^2$ as compared with (4.3). When $p > 1$, the cost of the update (2.7) also increases by a factor of 6 using complex arithmetic. Hence, the exact flop count of `KPShiftSolve` lies between what is specified in (4.3) and the bounds given above when complex arithmetic is used for the entire process.

If $n_1 = \cdots = n_p \equiv n$ in (4.3), then it can be shown that

$$\nu_p = \frac{1 + p + p^2}{2} n^{p+1} = \frac{1 + p + p^2}{2} Nn.$$

Things are even more complicated if the $n_i$ vary. For example, if the $T^{(i)}$ are triangular and real, then having $n_1 \leq \cdots \leq n_p$ is more advantageous than having $n_1 \geq \cdots \geq n_p$ to reduce the number of recursive calls. For quasi-triangular $T^{(i)}$, the flop count, vectorization properties, and recursion overheads depend upon the size ordering and the number of 2-by-2 bumps in each $T^{(i)}$.

**5. Implementation issues and performance.** `KPShiftSolve` can be made more efficient in two ways. First, the $T^{(i)}$ should be sorted so that $T^{(i+1)}$ has fewer 2-by-2 bumps than $T^{(i)}$, $i = 1{:}p - 1$, so there are fewer recursive calls. This can be accomplished via the perfect shuffle explained in section 2 and reduces the overall flop count. Second, instead of computing the real Schur form of $T^{(1)}$, we need only compute the cheaper Hessenberg decomposition. To appreciate why this is sufficient consider the $p = 2$ example at the start of section 3. If $G$ is upper Hessenberg then the linear systems with coefficient matrices $f_{ii}G - \lambda I$ that arise during the back-substitution process (3.2) are also upper Hessenberg. Hessenberg systems can be solved as quickly as quasi-triangular systems [1, p. 155].

MATLAB codes for the algorithms discussed in this paper are available at

http://www.math.jmu.edu/∼carlam/.

The reader interested in details, especially as they concern the recursion, should study the codes directly.

With respect to the roundoff properties of the algorithm, the computed solution $\hat{x}$ can be shown to solve

$$\left(A^{(p)} \otimes \cdots \otimes A^{(1)} - \lambda I_N + E\right) \hat{x} = b, \tag{5.1}$$

```
function $y = \texttt{KPShiftSolve}(T, n, c, \lambda, \alpha)$
% $T$ is a length $p$ cell array and $n = (n_1, \ldots, n_p)$. Assume that the
% $i$th element of $T$ is the upper quasi-triangular matrix $T_i$.
% Set $N = n_1 \cdots n_p$. $\alpha$ is a scalar or a 2-by-2 matrix. If $\lambda$ is a real scalar
% and $c \in \mathbb{R}^N$, then $y \in \mathbb{R}^N$ solves $(\alpha \otimes T_p \otimes \cdots \otimes T_1 - \lambda I)y = c$ assuming that
% the system is nonsingular. $\alpha = 1$ is the default value when not specified.

$p = \text{length}(n);$        $N = \text{prod}(n);$
if $\alpha \in \mathbb{R}$
    $T_p = \alpha T_p$
    if $p == 1$
        Solve $(T_1 - \lambda I_{n_1})y = c$ for $y$.
    else
        $y = \text{zeros}(N, 1);$        $m_p = N/n_p;$        $i = n_p;$
        while $(i \geq 1)$
            if $i > 1$ & $T_p(i, i-1) \neq 0$        ($T_p$ has a 2-by-2 bump)
                $idx = 1 + (i-2)m_p{:}\, im_p$
                $y(idx) = \texttt{KPShiftSolve}(T, n(1{:}p-1), c(idx), \lambda, T_p(i-1{:}i, i-1{:}i))$
                $z_1 = (T_{p-1} \otimes \cdots \otimes T_1) \cdot y(idx(1){:}(i-1)m_p)$              (Invoke (2.7))
                $z_2 = (T_{p-1} \otimes \cdots \otimes T_1) \cdot y(1 + (i-1)m_p{:}idx(end))$      (Invoke (2.7))
                for $j = 1{:}i-2$
                    $jdx = 1 + (j-1)m_p{:}\, jm_p$
                    $c(jdx) = c(jdx) - T_p(j, i-1)z_1 - T_p(j, i)z_2$
                end
                $i = i - 2$
            else        ($T_p$ does not have a 2-by-2 bump)
                $idx = 1 + (i-1)m_p{:}\, im_p$
                $y(idx) = \texttt{KPShiftSolve}(T, n(1{:}p-1), c(idx), \lambda, T_p(i, i))$
                $z = (T_{p-1} \otimes \cdots \otimes T_1)\, y(idx)$        (Invoke (2.7))
                for $j = 1{:}i-1$
                    $jdx = 1 + (j-1)m_p{:}\, jm_p$
                    $c(jdx) = c(jdx) - T_p(j, i)z$
                end
                $i = i - 1$
            end
        end
else $(\alpha \in \mathbb{R}^{2\times 2})$
    Compute $Q$ unitary, $S$ upper triangular so that $Q^H \alpha Q = S$
    $d = (Q^H \otimes I)c;$        $T_{p+1} = S;$        $n_{p+1} = 2;$
    $z = \texttt{KPShiftSolve}(T, n, d, \lambda, 1)$
    $y = (Q \otimes I)z;$        $y = \texttt{real}(y);$
end
```

FIG. 1. *Pseudo*-MATLAB *code for* `KPShiftSolve`.

where for any $p$-norm

$$\| E \| \approx \mathbf{u}\Big(\| A^{(p)} \| \cdots \| A^{(1)} \| + |\lambda|\Big) = \mathbf{u}\Big(\| A^{(p)} \otimes \cdots \otimes A^{(1)} \| + |\lambda|\Big) \tag{5.2}$$

and $\mathbf{u}$ is the unit roundoff. See the appendix for details.

We make a three comments related to (5.2). First, the explicit formation of the coefficient matrix involves rounding errors of the same magnitude as $\| E \|$. Second, as with any shifted, nonsymmetric linear system, there is not much we can say about the forward stability in $\hat{x}$ because the connection between the condition and the shift parameter is nontrivial. Finally, if $\kappa_p(\cdot)$ denotes the $p$-norm condition, then $\kappa_p(A^{(p)} \otimes \cdots \otimes A^{(1)}) = \kappa_p(A^{(p)}) \cdots \kappa_p(A^{(1)})$. Thus, modest ill-conditioning among the $A^{(i)}$ compounds to severe ill-conditioning in the Kronecker product.

## 6. Conclusion.

We have presented an algorithm that solves the shifted system

$$\Big(A^{(p)} \otimes \cdots \otimes A^{(1)} - \lambda I_N\Big)\, x = b,$$

where $A^{(i)} \in \mathbb{R}^{n_i \times n_i}$ for $i = 1{:}p$ and $N = n_1 \cdots n_p$. Our algorithm involves taking the real Schur decompositions to convert this system to a quasi-triangular system and uses a recursive block back-substitution procedure (`KPShiftSolve`). When a 2-by-2 bump is encountered in the leading coefficient matrix, the complex Schur decomposition is computed by the 2-by-2 matrix. This is faster than computing the complex Schur decompositions from the start. The error associated with our algorithm is no worse than the method of actually forming the Kronecker product and using standard back-substitution.

## Appendix. Error analysis.

In this appendix, we establish (5.1) and (5.2). Recall that the first step in our algorithm is to compute the (real) Schur decompositions of each $A^{(i)}$. Because of the results in [3], it suffices to show that if $\hat{y}$ is produced by `KPShiftSolve` then

$$(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I + \Delta T)\hat{y} = c, \tag{A.1}$$

$$\| \Delta T \| \le \delta_T(||T^{(p)}|| \cdots ||T^{(1)}|| + |\lambda|), \tag{A.2}$$

where $\delta_T$ is a modest multiple of the unit roundoff $\mathbf{u}$ and $T^{(i)}$ is either triangular or quasi-triangular for $i = 1{:}p$. To establish these results we first say something about the case $p = 2$. As in [3], we adopt the convention that all the $\delta$'s below are $O(\mathbf{u})$ in magnitude. In addition, the floating point result of a matrix calculation is indicated by fl$(\cdot)$ and we use "hat" notation to represent computed quantities.

Lemma A.1. [illegible] $F \in \mathbb{R}^{m\times m}$ [illegible] $G \in \mathbb{R}^{n\times n}$ [illegible] $c \in \mathbb{R}^{mn}$ [illegible] $\lambda \in \mathbb{R}$ [illegible] $\hat{y}$ [illegible] `KPShiftSolve` [illegible] $(F \otimes G - \lambda I)y = c$ [illegible] $E \in \mathbb{R}^{mn\times mn}$ [illegible]

$$(F \otimes G - \lambda I + E)\hat{y} = c, \tag{A.3}$$

$$\| E \| \le \delta_E(\| F \| \| G \| + |\lambda|). \tag{A.4}$$

[illegible] The proof is by induction on the dimension of $F$. If $m = 1$ then we solve $(f_{11}G - \lambda I_n)y = c$. This involves first forming $M = f_{11}G - \lambda I_n$ and then solving

$My = c$ using back-substitution. Accounting for the rounding error associated with forming $M$, there exists $H_1$ such that

$$\hat{M} = \mathrm{fl}(M) = f_{11}G - \lambda I + H_1, \qquad \| H_1 \| \le \delta_1(|f_{11}| \cdot \| G \| + |\lambda|). \tag{A.5}$$

Next, the computed solution to the triangular system satisfies

$$(\hat{M} + H_2)\hat{y} = c, \tag{A.6}$$

$$\| H_2 \| \le \delta_2 \| \hat{M} \| \le \delta_3(|f_{11}| \cdot \| G \| + |\lambda|), \tag{A.7}$$

where $\delta_3 = \delta_1 + \delta_2$ (see [1, p. 89]). Combining (A.5)–(A.7) and setting $E = H_1 + H_2$ complete the proof when $m = 1$.

Now suppose (A.3), (A.4) hold for all $k < m$. Partition the system as

$$\begin{bmatrix} f_{11}G - \lambda I & F_{12} \otimes G \\ 0 & F_{22} \otimes G - \lambda I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix},$$

where $F_{12} = F(1, 2{:}m)$, $F_{22} = F(2{:}m, 2{:}m)$, and $y_1, y_2, c_1, c_2$ are appropriate blockings of $y$ and $c$, respectively. By induction, $\hat{y}_2$ solves

$$(F_{22} \otimes G - \lambda I + E_{22})\hat{y}_2 = c_2$$

with

$$\| E_{22} \| \le \delta_4(\| F_{22} \| \| G \| + |\lambda|). \tag{A.8}$$

The next step is to solve for $y_1$ with

$$(f_{11}G - \lambda I)y_1 = \mathrm{fl}(c_1 - (F_{12} \otimes G)\hat{y}_2).$$

The computations associated with the update $d_1 = c_1 - (F_{12} \otimes G)\hat{y}_2$ satisfy

$$\hat{d}_1 = \mathrm{fl}(d_1) = c_1 - (F_{12} \otimes G + E_{12})\hat{y}_2 + \Delta c,$$

where

$$\| E_{12} \| \le \delta_5 \| F_{12} \| \| G \|, \tag{A.9}$$

$$\| \Delta c \| \le \delta_6(\| c_1 \| + \| F_{12} \| \| G \| \| \hat{y}_2 \|). \tag{A.10}$$

Finally, we form $M = f_{11}G - \lambda I$ and solve $My_1 = d_1$ using back-substitution. Forming $M$ gives

$$\hat{M} = \mathrm{fl}(M) = f_{11}G - \lambda I + \Delta_1, \tag{A.11}$$

$$\| \Delta_1 \| \le \delta_7(|f_{11}| \cdot \| G \| + |\lambda|). \tag{A.12}$$

Then the computed solution to the triangular system $\hat{M}y_1 = \hat{d}_1$ satisfies

$$(\hat{M} + \Delta_2)\hat{y}_1 = \hat{d}_1$$

with

$$\| \Delta_2 \| \le \delta_8 \| \hat{M} \| \le \delta_9(|f_{11} \cdot \| G \| + |\lambda|). \tag{A.13}$$

Let $E_{11} = \Delta_1 + \Delta_2$ and set

$$E = \begin{bmatrix} E_{11} & E_{12} \\ 0 & E_{22} \end{bmatrix}.$$

The proof follows from (A.8), (A.9), (A.12), (A.13), and by setting $\delta_E = \delta_4 + \delta_5 + \delta_7 + \delta_9$. □

Next, we address the error associated with computations when a 2-by-2 bump is encountered. For simplicity, we show this for $p = 2$.

Lemma A.2. [illegible] $\alpha$ [illegible] $2 \times 2$ [illegible] $T$ [illegible] $n \times n$ [illegible] $m = n\dim(\alpha)$ $c \in \mathbb{R}^m$ [illegible] $\lambda \in \mathbb{R}$ [illegible] $\hat{y}$ [illegible] `KPShiftSolve` [illegible] $(\alpha \otimes T - \lambda I)y = c$ [illegible] $E \in \mathbb{R}^{m\times m}$ [illegible]

$$(\alpha \otimes T - \lambda I_m + E)\hat{y} = c, \tag{A.14}$$

$$\| E \| \leq \delta_E(\| \alpha \| \| T \| + |\lambda|). \tag{A.15}$$

[illegible]. If $\alpha$ is $1\times 1$, then the proof is completed by using Lemma A.1. If $\alpha$ is $2\times 2$, then `KPShiftSolve` computes the complex Schur decomposition of $\alpha$ and then solves a block upper triangular system. Specifically, `KPShiftSolve` performs the following four steps:

1. Compute the complex Schur decomposition $Q^H \alpha Q = S$.
2. Form $d = (Q^H \otimes I)c$.
3. Solve the system $(S \otimes T - \lambda I)z = d$ for $z$ using `KPShiftSolve`.
4. Set $y = (Q \otimes I)z$.

These steps are analogous to solving a linear system using the complex Schur decomposition. The Kronecker product structure does not affect the result and can be shown using the methods found in [3] and Lemma A.1. □

Now that we have dealt with the error associated with solving the system when a 2-by-2 bump is encountered, we are ready to establish the error results for a Kronecker product of quasi-triangular matrices. In the next lemma, we present the results for $p = 2$.

Lemma A.3. [illegible] $F$ [illegible] $G$ [illegible] $m \times m$ [illegible] $n \times n$ [illegible] $\lambda \in \mathbb{R}$ [illegible] $b \in \mathbb{R}^{mn}$ [illegible] `KPShiftSolve` [illegible] $(F \otimes G - \lambda I_{mn})y = c$ [illegible] $E \in \mathbb{R}^{mn\times mn}$ [illegible] $\hat{y}$ [illegible]

$$(F \otimes G - \lambda I + E)\hat{y} = c, \tag{A.16}$$

$$\| E \| \leq \delta_E(\| F \| \| G \| + |\lambda|). \tag{A.17}$$

[illegible]. The proof is similar to the proof of Lemma A.1 and is completed by induction on the dimension of $F$. Lemma A.2 is used when a 2-by-2 bump is encountered in $F$. □

We are now ready to establish the final result for general $p$. The following lemma establishes (A.1) and (A.2).

Lemma A.4. [illegible] $T^{(1)}, \ldots, T^{(p)}$ [illegible] $n_i \times n_i$ [illegible] $i = 1, \ldots, p$ [illegible] $\lambda \in \mathbb{R}$ [illegible] $b \in \mathbb{R}^{n_1 \cdots n_p}$ [illegible] `KPShiftSolve` [illegible] $(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I)y = c$ [illegible] $\Delta T \in \mathbb{R}^{N\times N}$ [illegible] $N = n_1 \cdots n_p$ [illegible]

[illegible] $\hat{y}$ [illegible]

$$(T^{(p)} \otimes \cdots \otimes T^{(1)} - \lambda I + \Delta T)\hat{y} = c,$$

$$\| \Delta T \| \leq \delta_T(||T^{(p)}|| \cdots ||T^{(1)}|| + |\lambda|).$$

[illegible]. We prove this by induction on $p$. Lemma A.3 proves the base case when $p = 2$. Now assume Lemma A.4 holds for $p - 1$. To show this is true for $p$, we use induction on $\dim(A^{(p)})$. The proof follows by following the methods similar to the proof of Lemma A.1 when $p = 2$. The proof now easily follows from Lemma A.3 by setting $G = T^{(p-1)} \otimes \cdots \otimes T^{(1)}$. ☐

## REFERENCES

[1] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[2] I. Jonsson and B. Kågström, *Recursive blocked algorithm for solving triangular systems.* II. *Two-sided and generalized Sylvester and Lyapunov matrix equations*, ACM Trans. Math. Software, 28 (2002), pp. 416–435.

[3] C. D. Moravitz Martin and C. F. Van Loan, *Solving real linear systems with the complex Schur decomposition*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 177–183.

[4] C. F. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, Front. Appl. Math. 10, SIAM, Philadelphia, 1992.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 199–227

# MULTISHIFT VARIANTS OF THE QZ ALGORITHM WITH AGGRESSIVE EARLY DEFLATION[*]

BO KÅGSTRÖM[†] AND DANIEL KRESSNER[†‡]

**Abstract.** New variants of the QZ algorithm for solving the generalized eigenvalue problem are proposed. An extension of the small-bulge multishift QR algorithm is developed, which chases chains of many small bulges instead of only one bulge in each QZ iteration. This allows the effective use of level 3 BLAS operations, which in turn can provide efficient utilization of high performance computing systems with deep memory hierarchies. Moreover, an extension of the aggressive early deflation strategy is proposed, which can identify and deflate converged eigenvalues long before classic deflation strategies would. Consequently, the number of overall QZ iterations needed until convergence is considerably reduced. As a third ingredient, we reconsider the deflation of infinite eigenvalues and present a new deflation algorithm, which is particularly effective in the presence of a large number of infinite eigenvalues. Combining all these developments, our implementation significantly improves existing implementations of the QZ algorithm. This is demonstrated by numerical experiments with random matrix pairs as well as with matrix pairs arising from various applications.

**Key words.** generalized eigenvalue problem, generalized Schur form, QZ algorithm, multishifts, aggressive early deflation, blocked algorithms

**AMS subject classifications.** 65F15, 15A18, 15A22, 47A75

**DOI.** 10.1137/05064521X

## 1. Introduction.

The QZ algorithm is a numerically backward stable method for computing generalized eigenvalues and deflating subspaces of small- to medium-sized regular matrix pairs $(A, B)$ with $A, B \in \mathbb{R}^{n\times n}$. It goes back to Moler and Stewart in 1973 [37] and underwent only a few modifications during the following 25 years, notably through works by Ward [47, 48], Kaufman [29], and Dackland and Kågström [12]. Nonorthogonal variants of the QZ algorithm include the LZ algorithm by Kaufman [28] and the AB algorithm for pencils by Kublanovskaya [34].

The purpose of the QZ algorithm is to compute a [illegible] of $(A, B)$, i.e., orthogonal matrices $Q$ and $Z$ so that $S = Q^TAZ$ is quasi-upper triangular with $1 \times 1$ and $2 \times 2$ blocks on the diagonal, while the matrix $T = Q^TBZ$ is upper triangular. This decomposition provides almost everything needed to solve the generalized nonsymmetric eigenvalue problem (GNEP). [illegible], defined as root pairs $(\alpha, \beta)$ of the bivariate polynomial $\det(\beta A - \alpha B)$, can be directly computed from the diagonal blocks of $S$ and $T$, although some care must be taken to implement this computation in a safe manner; see [37, 45]. Moreover, the leading $k$ columns of the orthogonal matrices $Z$ and $Q$ span a pair of [illegible] [40] if the $(k+1, k)$ subdiagonal entry of the matrix $S$ vanishes. A reordering of the diagonal blocks of $S$ and $T$ can be used to compute other deflating subspaces; see [26, 25, 44].

The eigenvalues of $(A, B)$ are read off from $(S, T)$ as follows. The $2 \times 2$ diagonal blocks correspond to pairs of complex conjugate eigenvalues. The real eigenvalues

---

[*] Received by the editors November 15, 2005; accepted for publication (in revised form) by P. Benner August 21, 2006; published electronically December 21, 2006. Supported by the DFG Emmy Noether fellowship KR 2950/1-1 and by the *Swedish Research Council* under grant VR 621-2001-3284 and by the *Swedish Foundation for Strategic Research* under grant A3 02:128. This research was conducted using the resources of the High Performance Computing Center North (HPC2N).

http://www.siam.org/journals/simax/29-1/64521.html

[†] Department of Computing Science and HPC2N, Umeå University, S-901 87 Umeå, Sweden (bokg@cs.umu.se, kressner@cs.umu.se).

[‡] Department of Mathematics, Bijenička 30, 10000 Zagreb, Croatia (kressner@math.hr).

are given in pairs $(s_{ii}, t_{ii})$ corresponding to the $1 \times 1$ diagonal blocks of $(S, T)$. The finite eigenvalues are $s_{ii}/t_{ii}$, where $t_{ii} \neq 0$. An infinite eigenvalue is represented as $(s_{ii}, 0)$ with $s_{ii} \neq 0$. If $(s_{ii}, t_{ii}) \neq (0, 0)$ for all $i$, then $(A, B)$ is a [illegible], or equivalently $\beta A - \alpha B$ is a regular matrix pencil. Otherwise, the matrix pair is [illegible] and at least one $(s_{ii}, t_{ii})$ equals $(0, 0)$. These situations need extra caution, and so-called staircase-type algorithms can be used for identifying singular cases by computing a generalized upper triangular (GUPTRI) form of $(A, B)$ (e.g., see Demmel and Kågström [13, 14]).

Three ingredients make the QZ algorithm work effectively. First, the matrix pair $(A, B)$ is reduced to Hessenberg-triangular form; i.e., orthogonal matrices $Q$ and $Z$ are computed so that $H = Q^T AZ$ is upper Hessenberg and $T = Q^T BZ$ is upper triangular. Second, a sequence of so-called implicit shifted QZ iterations is applied to $(H, T)$ in order to bring $H$ closer to (block) upper triangular form while preserving the Hessenberg-triangular form of $(H, T)$. Each of these iterations can be seen as chasing a pair of bulges from the top left to the bottom right corners along the subdiagonals of $H$ and $T$, a point of view that has been emphasized by Watkins and Elsner [49]. The third ingredient is deflation, which aims at splitting the computation of the generalized Schur form $(S, T)$ into smaller subproblems. This paper describes improvements for the latter two ingredients, QZ iterations, and deflations.

Inspired by the works of Braman, Byers, and Mathias [7] and Lang [36] for the QR algorithm, we propose multishift QZ iterations that chase a tightly coupled chain of bulge pairs instead of only one bulge pair per iteration. This allows the effective use of level 3 BLAS operations [15, 23, 24] during the bulge chasing process, which in turn can provide efficient utilization of today's high performance computing systems with deep memory hierarchies. Tightly coupled bulge chasing has also successfully been used in the reduction of a matrix pair $(H_r, T)$ in block Hessenberg-triangular form, where $H_r$ has $r$ subdiagonals, to Hessenberg-triangular form $(H, T)$ [12].

Recently, Braman, Byers, and Mathias [6] also presented a new, advanced deflation strategy, the so-called aggressive early deflation. Combining this deflation strategy with multishift QR iterations leads to a variant of the QR algorithm, which may, for sufficiently large matrices, require less than 10% of the computing time needed by the LAPACK [2] implementation. We will show that this deflation strategy can be extended to the QZ algorithm, resulting in similar time savings.

A (nearly) singular matrix $B$ often implies that the triangular matrix $T$ of the corresponding Hessenberg-triangular form has one or more diagonal entries close to zero. Each of these diagonal entries admits the deflation of an infinite eigenvalue. Some applications, such as semidiscretized Stokes equations [42], lead to matrix pairs that have a large number of infinite eigenvalues. Consequently, a substantial amount of computational work in the QZ algorithm is spent deflating these eigenvalues. We will provide a discussion on this matter including preprocessing techniques, and we propose windowing techniques that lead to more efficient algorithms for deflating infinite eigenvalues within the QZ algorithm. This approach is conceptually close to blocked algorithms for reordering eigenvalues in standard and generalized Schur forms [32].

The rest of this paper is organized as follows. In section 2, we review and extend conventional multishift QZ iterations and provide some new insight into their numerical backward stability. Multishift variants that are based on chasing a tightly coupled chain of bulge pairs are described in section 3. In section 4, a thorough discussion on dealing with infinite eigenvalues is presented that includes preprocessing and efficient methods for deflating such eigenvalues within the QZ algorithm. Aggressive early deflation for the QZ algorithm and its connection to the distance of uncontrollability for descriptor systems are studied in section 6. Computational experiments, presented in section 7, demonstrate the effectiveness of our newly developed multishift QZ algorithm with advanced deflation techniques. Finally, some concluding remarks are summarized in section 8.

**2. Conventional multishift QZ iterations.** Throughout the rest of this paper we assume that the matrix pair under consideration, which will be denoted by $(H, T)$, is already in Hessenberg-triangular form. Efficient algorithms for reducing a given matrix pair to this form can be found in [12, 31]. For the moment, we also assume that $(H, T)$ is an [illegible]; i.e., all subdiagonal entries of $H$ as well as all diagonal entries of $T$ are different from zero. The latter condition implies that only finite eigenvalues are considered.

A QZ iteration relies on a fortunate choice of $m$ shifts (or shift pairs) $(\mu_1, \nu_1)$, $(\mu_2, \nu_2), \dots, (\mu_m, \nu_m)$ with $\mu_i \in \mathbb{C}$ and $\nu_i \in \mathbb{R}$, giving rise to the [illegible]

$$p(HT^{-1}) = (\nu_1 HT^{-1} - \mu_1 I_n)(\nu_2 HT^{-1} - \mu_2 I_n) \cdots (\nu_m HT^{-1} - \mu_m I_n). \tag{2.1}$$

If $x$ denotes the first column of this matrix polynomial, then the first step of an implicit shifted QZ iteration consists of choosing an orthogonal matrix $Q_1$ such that $Q_1^T x$ is a multiple of the first unit vector $e_1$. The rest of the QZ iteration consists of reducing the updated matrix pair $(Q_1^T H, Q_1^T T)$ back to Hessenberg-triangular form, without modifying the first rows of $Q_1^T H$ and $Q_1^T T$ by transformations from the left.

In the original formulation of the QZ algorithm [37], this reduction to Hessenberg-triangular form was described for $m \leq 2$, based on combinations of Givens rotations and Householder matrices. This approach has the negative side-effect that one QZ iteration with $m = 2$ shifts requires more [illegible] (floating point operations) than two QZ iterations with $m = 1$ shift. Partly avoiding this increase of flops, Ward [47] proposed the so-called [illegible] which uses $m = 1$ for real shifts and $m = 2$ for complex conjugate pairs of shifts. Later on, Watkins and Elsner [49] proposed a variant solely based on Householder matrices which requires roughly 27% fewer flops than the original formulation and may employ an arbitrary number $m$ of shifts. This variant is currently implemented in the LAPACK subroutine `DHGEQZ`. A curiosity of this subroutine is that it still uses Ward's combination shift strategy despite the fact that two single shift QZ iterations now require roughly 9% more flops than one double shift iteration.

**2.1. Householder-based variants.** In the following, we describe the Householder-based variant by Watkins and Elsner in more detail. To simplify the notation, we make use of the following convention.

DEFINITION 2.1. [illegible] $n - j$ [illegible] $x \in \mathbb{R}^n$ [illegible] $j - 1$ [illegible] $\mathcal{H}_j(x)$

Let us illustrate the first few steps of an implicit QZ iteration for $n = 6, m = 2$. First, a Householder matrix $\mathcal{H}_1(x)$ is used to map $x$, the first column of the shift polynomial defined in (2.1), to a multiple of $e_1$. Note that only the leading three elements of $x$ are nonzero. Hence, if $\mathcal{H}_1(x)$ is applied from the left to $H$ and $T$, only the first three rows (denoted by the symbols $\hat{h}$ and $\hat{t}$ below) are affected while the remaining rows stay unchanged (denoted by $h$ and $t$):

$$(H,T) \leftarrow \left( \begin{bmatrix} \hat h & \hat h & \hat h & \hat h & \hat h & \hat h \\ \hat h & \hat h & \hat h & \hat h & \hat h & \hat h \\ \hat h & \hat h & \hat h & \hat h & \hat h & \hat h \\ 0 & 0 & h & h & h & h \\ 0 & 0 & 0 & h & h & h \\ 0 & 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} \hat t & \hat t & \hat t & \hat t & \hat t & \hat t \\ \hat t & \hat t & \hat t & \hat t & \hat t & \hat t \\ \hat t & \hat t & \hat t & \hat t & \hat t & \hat t \\ 0 & 0 & 0 & t & t & t \\ 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & t \end{bmatrix} \right). \tag{2.2}$$

Next, to avoid further fill-in in the factor $T$, the newly introduced entries $(2,1)$ and $(3,1)$ must be eliminated. Recall that we are not allowed to change the first row of $T$ by applying a transformation from the left. However, it is still possible to achieve these eliminations by applying a Householder matrix using the following simple fact.

LEMMA 2.2 (see [49]). [illegible] $T \in \mathbb{R}^{n\times n}$ [illegible] $T\mathcal{H}_1(T^{-1}e_1)$ [illegible] $e_1$

Applying a Householder matrix from the [illegible] to eliminate several elements in one [illegible] (instead of one row) is somewhat opposite to their standard use. This motivates us to call such a matrix an [illegible]. Applying $\mathcal{H}_1(T^{-1}e_1)$ from the right yields the following diagram:

$$(H,T) \leftarrow \left( \begin{bmatrix} \hat h & \hat h & \hat h & h & h & h \\ \hat h_b & \hat h_b & \hat h_b & h & h & h \\ \hat h_b & \hat h_b & \hat h_b & h & h & h \\ \hat h_b & \hat h_b & \hat h_b & h & h & h \\ 0 & 0 & 0 & h & h & h \\ 0 & 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} \hat t & \hat t & \hat t & t & t & t \\ \hat 0_b & \hat t_b & \hat t_b & t & t & t \\ \hat 0_b & \hat t_b & \hat t_b & t & t & t \\ 0_b & 0_b & 0_b & t & t & t \\ 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & t \end{bmatrix} \right). \tag{2.3}$$

Here, we have used the subscript $b$ to designate entries that belong to the so-called [illegible]. The rest of the QZ iteration can be seen as pushing this bulge pair along the subdiagonals down to the bottom right corners until it vanishes. The next two steps consist of applying the Householder matrix $\mathcal{H}_2(He_1)$ from the left and the opposite Householder matrix $\mathcal{H}_2(T^{-1}e_2)$ from the right:

$$(H,T) \leftarrow \left( \begin{bmatrix} h & h & h & h & h & h \\ \hat h & \hat h & \hat h & \hat h & \hat h & \hat h \\ \hat 0 & \hat h & \hat h & \hat h & \hat h & \hat h \\ \hat 0 & \hat h & \hat h & \hat h & \hat h & \hat h \\ 0 & 0 & 0 & h & h & h \\ 0 & 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} t & t & t & t & t & t \\ 0 & \hat t & \hat t & \hat t & \hat t & \hat t \\ 0 & \hat t & \hat t & \hat t & \hat t & \hat t \\ 0 & \hat t & \hat t & \hat t & \hat t & \hat t \\ 0 & 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & 0 & t \end{bmatrix} \right),$$

$$(H,T) \leftarrow \left( \begin{bmatrix} h & \hat h & \hat h & \hat h & h & h \\ h & \hat h & \hat h & \hat h & h & h \\ 0 & \hat h_b & \hat h_b & \hat h_b & h & h \\ 0 & \hat h_b & \hat h_b & \hat h_b & h & h \\ 0 & \hat h_b & \hat h_b & \hat h_b & h & h \\ 0 & 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} t & \hat t & \hat t & \hat t & t & t \\ 0 & \hat t & \hat t & \hat t & t & t \\ 0 & \hat 0_b & \hat t_b & \hat t_b & t & t \\ 0 & \hat 0_b & \hat t_b & \hat t_b & t & t \\ 0 & 0_b & 0_b & 0_b & t & t \\ 0 & 0 & 0 & 0 & 0 & t \end{bmatrix} \right). \tag{2.4}$$

For general $m$ and $n$, the implicit shifted QZ iteration based on (opposite) Householder matrices is described in Algorithm 1. Here, the [illegible] $A(i_1 : i_2, j_1 : j_2)$ is used to designate the submatrix of a matrix $A$ defined by rows $i_1$ through $i_2$ and columns $j_1$ through $j_2$.

Note that the shifts employed in Algorithm 1 are based on the generalized eigenvalues of the bottom right $m \times m$ submatrix pair, a choice which is sometimes called [illegible] and which ensures quadratic local convergence [49]. If $m \ll n$, a proper implementation of this algorithm requires $2(4m+3)n^2 + \mathcal{O}(n)$ flops for updating $H$ and $T$. In addition, $(4m+3)n^2 + \mathcal{O}(n)$ flops are required for updating each of the orthogonal factors $Q$ and $Z$.

**Algorithm 1** Implicit shifted QZ iteration based on Householder matrices

**Input:** An $n \times n$ matrix pair $(H, T)$ in unreduced Hessenberg-triangular form, an integer $m \in [2, n]$.

**Output:** Orthogonal matrices $Q, Z \in \mathbb{R}^{n\times n}$ so that $Q^T(H,T)Z$ is the Hessenberg-triangular matrix pair obtained after applying a QZ iteration with $m$ shifts. The matrix pair $(H, T)$ is overwritten by $Q^T(H,T)Z$.

```
Compute $(\mu_1,\nu_1), (\mu_2,\nu_2), \dots, (\mu_m,\nu_m)$ as generalized eigenvalues of the matrix pair
    $(H(n-m+1:n, n-m+1:n), T(n-m+1:n, n-m+1:n))$.
Set $x = ((\nu_1 HT^{-1} - \mu_1 I_n)(\nu_2 HT^{-1} - \mu_2 I_n)\cdots(\nu_m HT^{-1} - \mu_m I_n))e_1$.
$(H,T) \leftarrow \mathcal{H}_1(x)\cdot(H,T)$
$Q \leftarrow \mathcal{H}_1(x)$,    $Z \leftarrow \mathcal{H}_1(T^{-1}e_1)$
$(H,T) \leftarrow (H,T)\cdot Z$
for $j \leftarrow 1,2,\dots,n-2$ do
  $\tilde{Q} \leftarrow \mathcal{H}_{j+1}(He_j)$
  $(H,T) \leftarrow \tilde{Q}\cdot(H,T)$
  $Q \leftarrow Q\tilde{Q}$
  $\tilde{Z} \leftarrow \mathcal{H}_{j+1}(T^{-1}e_{j+1})$
  $(H,T) \leftarrow (H,T)\cdot\tilde{Z}$
  $Z \leftarrow Z\tilde{Z}$
end for
```

**2.2. Error analysis of opposite Householder matrices.** Some authors have raised concerns that the use of opposite Householder matrices could introduce numerical instabilities in the QZ algorithm; see, e.g., [12, p. 444]. Such instabilities could arise if some entries that should be zero after the application of an opposite Householder matrix are nonnegligible in finite-precision arithmetic. In the following, we provide a brief error analysis showing that such an event may not occur if some care is taken.

Without loss of generality, we can restrict the analysis to an opposite Householder matrix of the form $\mathcal{H}_1(T^{-1}e_1)$ for some nonsingular matrix $T \in \mathbb{R}^{n\times n}$. Although an ill-conditioned $T$ may severely affect the data representing $\mathcal{H}_1(T^{-1}e_1)$, it has almost no effect on the purpose of $\mathcal{H}_1(T^{-1}e_1)$, which is the introduction of zero entries. To explain this, assume that a numerically backward stable method is employed to solve the linear system $Tx = e_1$, yielding a computed solution $\hat{x}$. This implies that $\hat{x}$ is the exact solution of a slightly perturbed system

$$(T+F)\hat{x} = e_1, \quad \|F\|_2 \le c_T\|T\|_2, \tag{2.5}$$

where $c_T$ is not much larger than the unit roundoff $\mathbf{u}$ [21]. Now, consider the Householder matrix $\mathcal{H}_1(\hat{x}) = I - \tilde{\beta}\tilde{v}\tilde{v}^T$, where $\tilde{\beta} \in \mathbb{R}, \tilde{v} \in \mathbb{R}^n$, such that $(I - \tilde{\beta}\tilde{v}\tilde{v}^T)\hat{x} = \tilde{\gamma}e_1$ for some scalar $\tilde{\gamma}$. The computation of the quantities $\tilde{\beta}, \tilde{v}$ defining $\mathcal{H}_1(\hat{x})$ is also subject to roundoff errors. Using standard computational methods, the computed quantities $\hat{v}, \hat{\beta}$ satisfy

$$|\hat{\beta} - \tilde{\beta}| \le c_\beta|\tilde{\beta}| \approx (4n+8)\mathbf{u}|\tilde{\beta}|, \quad \|\hat{v} - \tilde{v}\|_2 \le c_v\|\tilde{v}\|_2 \approx (n+2)\mathbf{u}\|\tilde{v}\|_2;$$

see [21, p. 365]. It follows that

$$\begin{aligned}\|T\cdot(I - \hat{\beta}\hat{v}\hat{v}^T)e_1 - 1/\tilde{\gamma}\cdot e_1\|_2 &\le \|T\cdot(I - \tilde{\beta}\tilde{v}\tilde{v}^T)e_1 - 1/\tilde{\gamma}\cdot e_1\|_2 \\ &\quad + (2c_\beta + 4c_v)\|T\|_2 + \mathcal{O}(\mathbf{u}^2) \\ &\le (c_T + 2c_\beta + 4c_v)\|T\|_2 + \mathcal{O}(\mathbf{u}^2).\end{aligned}$$

This shows that if $\hat{x}$ is computed by a backward stable method, then the last $n-1$ elements in the first column of $T(I - \beta\hat{v}\hat{v}^T)$ can be set to zero without spoiling the backward stability of the QZ algorithm.

In this paper, we favor the following method for constructing opposite Householder matrices. Let $T = RQ$ be an RQ decomposition; i.e., the matrix $R \in \mathbb{R}^{n\times n}$ is upper triangular and $Q \in \mathbb{R}^{n\times n}$ is orthogonal. If $T$ is invertible, then $Q^T e_1$ is a scalar multiple of $T^{-1}e_1$ implying that $\mathcal{H}_1(Q^T e_1)$ is an opposite Householder matrix. Even if $T$ is singular, it can be shown that the first column of $T \cdot \mathcal{H}_1(Q^T e_1)$ is mapped to a multiple of $e_1$:

$$\begin{aligned} T \cdot \mathcal{H}_1(Q^T e_1) = R \cdot [Q \cdot \mathcal{H}_1(Q^T e_1)] &= \begin{bmatrix} r_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \begin{bmatrix} \tilde{q}_{11} & 0 \\ 0 & \tilde{Q}_{22} \end{bmatrix} \\ &= \begin{bmatrix} r_{11}\tilde{q}_{11} & R_{12}\tilde{Q}_{22} \\ 0 & R_{22}\tilde{Q}_{22} \end{bmatrix}. \end{aligned}$$

RQ decompositions enjoy a favorable backward error analysis, and the constant $c_T$ in (2.5) can be bounded by roughly $n^2\mathbf{u}$; see, e.g., [21, Thm. 18.4].

**2.3. Bulge pairs and shift blurring.** Convergence in the implicit shifted QZ iteration typically becomes manifest in the bottom right corner of the matrix pair; often the $m$th-last subdiagonal entry of $H$ converges to zero. As a QZ iteration can be interpreted as chasing a bulge pair from the top left corner to the bottom right corner of $(H, T)$, the question arises how the information contained in the shifts is passed during this chasing process. Watkins [53] discovered a surprisingly simple relationship; the intended shifts are the finite eigenvalues of the bulge pairs.

To explain this in more detail, suppose that the implicit shifted QZ iteration with $m$ shifts, Algorithm 1, is applied to $(H, T) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ with $n > m$. As before, we assume that $(H, T)$ is in unreduced Hessenberg-triangular form but we do not assume that $T$ is nonsingular; only the part used for the shift computation (the trailing $m \times m$ principal submatrix of $T$) and the part involved in the introduction of the bulge pair (the leading $m \times m$ principal submatrix of $T$) are required to be nonsingular. Let $x$ be a multiple of the first column of the shift polynomial defined in (2.1). The [illegible] is the matrix pair $\big(B_0^{(H)}, B_0^{(T)}\big)$, where

$$B_0^{(H)} = [x(1:m+1), H(1:m+1:1:m)] = \begin{bmatrix} x_1 & h_{11} & \cdots & h_{1m} \\ x_2 & h_{21} & \ddots & \vdots \\ \vdots & & \ddots & h_{mm} \\ x_{m+1} & 0 & & h_{m+1,m} \end{bmatrix},$$

$$B_0^{(T)} = [0, T(1:m+1:1:m)] = \begin{bmatrix} 0 & t_{11} & \cdots & t_{1m} \\ 0 & 0 & \ddots & \vdots \\ \vdots & & \ddots & t_{mm} \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

THEOREM 2.3 (see [53]). [illegible] $m \times m$ [illegible] $T$ [illegible] $(\sigma_1, 1), \dots, (\sigma_m, 1)$ [illegible] $\big(B_0^{(H)}, B_0^{(T)}\big)$

During the course of a QZ iteration, a bulge pair is first created at the top left corners and then chased down to the bottom right corners. Let $\big(H^{(j)}, T^{(j)}\big)$ denote the updated matrix pair $(H, T)$ obtained after the bulge pair has been chased $j-1$ steps, which amounts to applying $j-1$ loops of Algorithm 1. Then, the $j$[illegible] $\big(B_j^{(H)}, B_j^{(T)}\big)$ is given by

$$
\begin{array}{rcl}
B_j^{(H)} & = & H^{(j)}(j+1:j+m+1, j:j+m+1), \\
B_j^{(T)} & = & T^{(j)}(j+1:j+m+1, j:j+m+1),
\end{array}
\tag{2.6}
$$

which corresponds to the submatrices designated by the subscript $b$ in (2.3)–(2.4).

THEOREM 2.4 (see [53]). [illegible] $m$[illegible] $T$ [illegible] $\sigma_1, \ldots, \sigma_m$ [illegible] $j$[illegible] $\big(B_j^{(H)}, B_j^{(T)}\big)$

Note that the definition of a bulge pair is only possible for $j \leq n-m-1$, since otherwise (2.6) refers to entries outside of $H^{(j)}$ and $T^{(j)}$. This issue can be resolved by adding virtual rows and columns to the matrix pair $(H^{(j)}, T^{(j)})$; see [53]. Theorem 2.4 can be extended to the case $j > n-m-1$.

Early attempts to improve the performance of the QR algorithm focused on using shift polynomials of high degree [4], leading to medium-order Householder matrices during the QR iteration and enabling the efficient use of WY representations. This approach, however, has proved disappointing due to the fact that the convergence of such a large-bulge multishift QR algorithm is severely affected by roundoff errors [16]. This effect is caused by [illegible]: with increasing $m$ the eigenvalues of the bulge pairs, which should represent the shifts in exact arithmetic, often become extremely sensitive to perturbations [51, 52, 33]. Already for moderate $m$, say, $m \geq 15$, the shifts may be completely contaminated by roundoff errors during the bulge chasing process. Not surprisingly, we made similar observations in numerical experiments with implicit shifted QZ iterations, which also suffer from shift blurring.

## 3. Multishift QZ iterations based on tightly coupled tiny bulge pairs.

The trouble with shift blurring can be avoided by developing variants of the implicit shifted QZ algorithm that still rely on a large number of simultaneous shifts but chase several tiny bulge pairs instead of one large bulge pair. Such ideas have already been successfully applied to the QR algorithm; see, e.g., [7, 36] and the references therein. In this section, we describe an extension of the work by Braman, Byers, and Mathias [7] to the QZ algorithm.

For the purpose of describing this new tiny-bulge multishift QZ algorithm, let $m$ denote the number of simultaneous shifts to be used in each QZ iteration and let $n_s$ denote the number of shifts contained in each bulge pair. It is assumed that $m$ is an integer multiple of $n_s$. To avoid shift blurring phenomena we use tiny values for $n_s$, say, $n_s = 2$ or $n_s = 4$.

Our algorithm performs an implicit shifted QZ iteration with $m$ generalized Francis shifts to a Hessenberg-triangular matrix pair $(H, T)$ and consists of three stages, which are described in more detail below. First, a tightly coupled chain of $m/n_s$ bulge pairs is bulge-by-bulge introduced in the top left corners of $H$ and $T$. Second, the whole chain at once is chased down along the subdiagonal until the bottom bulge pair reaches the bottom right corners of $H$ and $T$. Finally, all bulge pairs are bulge-by-bulge chased off this corner.

### 3.1. Introducing a chain of bulge pairs.

The tiny-bulge multishift QZ algorithm begins with introducing $m/n_s$ bulge pairs in the top left corner of the matrix pair $(H, T)$. Every bulge pair contains a set of $n_s$ shifts. It is assumed that the $((m/n_s)(n_s+1)-1)$th leading principal submatrix of $T$ is nonsingular. The first

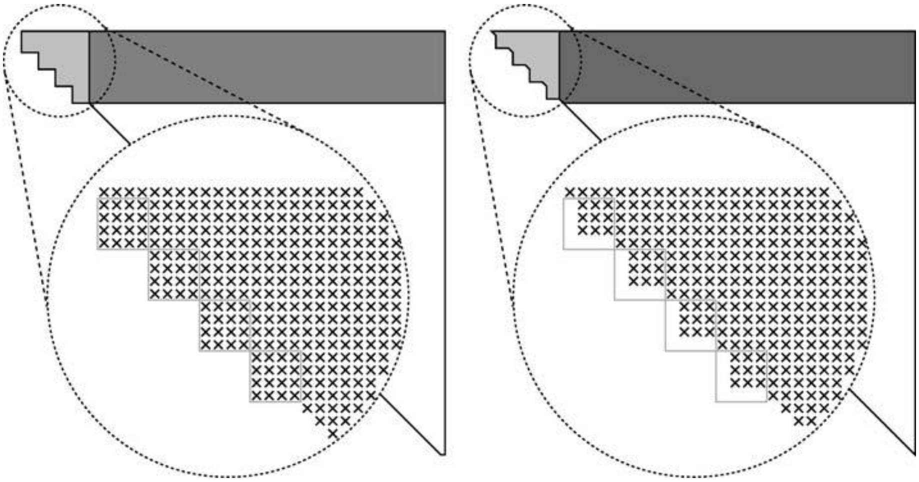


FIG. 3.1. *Introducing a chain of $m/n_s = 4$ tightly coupled bulge pairs, each of which contains $n_s = 3$ shifts.*

bulge pair is introduced by applying an implicit QZ iteration with $n_s$ shifts and interrupting the bulge chasing process as soon as the bottom right corner of the bulge in $H$ touches the $(p_h - 1, p_h)$ subdiagonal entry of $H$, where $p_h = (m/n_s)(n_s + 1) + 1$. The next bulge pair is chased until the bottom right corner of the bulge in $H$ touches the $(p_h - n_s - 2, p_h - n_s - 1)$ subdiagonal entry. This process is continued until all $m/n_s$ bulge pairs are introduced; see Figure 3.1. Note that only the submatrices marked light gray in Figure 3.1 must be updated during the bulge chasing process. To update the remaining parts (marked dark gray), all orthogonal transformations from the left are accumulated into a $p_h \times p_h$ matrix $U$ and applied in terms of general matrix-matrix multiply (GEMM) operations:

$$\begin{aligned} H(1 : p_h, (p_h + 1) : n) &\leftarrow U^T \cdot H(1 : p_h, (p_h + 1) : n), \\ T(1 : p_h, (p_h + 1) : n) &\leftarrow U^T \cdot T(1 : p_h, (p_h + 1) : n). \end{aligned}$$

**3.2. Chasing a chain of bulge pairs.** In each step of the tiny-bulge multishift QZ algorithm, the chain of bulge pairs is chased $k$ steps downward. Before the first step, this chain resides in columns/rows $p_l : p_h$ with $p_l = 1, p_h = (m/n_s)(n_s + 1) + 1$ as above. Before the next step, we have $p_l = 1 + k, p_h = (m/n_s)(n_s + 1) + 1 + k$, and so on.

The whole chain is chased in a bulge-by-bulge and bottom-to-top fashion. One such step is illustrated in Figure 3.2. Again, only the principal submatrices marked light gray in Figure 3.2 must be updated during the bulge chasing process. All transformations from the left and from the right are accumulated into orthogonal matrices $U$ and $V$, respectively. Then, GEMM operations can be used to update the rest of the matrix pair (marked dark gray in Figure 3.2):

$$\begin{aligned} H(p_l : p_h + k, (p_h + 1) : n) &\leftarrow U^T \cdot H(p_l : p_h + k, (p_h + 1) : n), \\ T(p_l : p_h + k, (p_h + 1) : n) &\leftarrow U^T \cdot T(p_l : p_h + k, (p_h + 1) : n), \\ H(1 : p_l - 1, p_l : p_h + k) &\leftarrow H(1 : p_l - 1, p_l : p_h + k) \cdot V, \\ T(1 : p_l - 1, p_l : p_h + k) &\leftarrow T(1 : p_l - 1, p_l : p_h + k) \cdot V. \end{aligned}$$

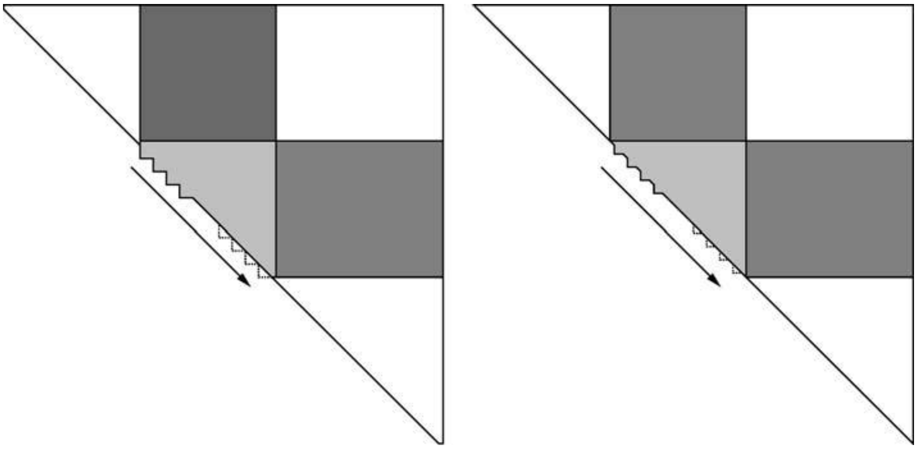


FIG. 3.2. *Chasing a chain of $m/n_s = 4$ tightly coupled bulge pairs.*

Note that both matrices, $U$ and $V$, have the following block structure:

$$\begin{array}{c} \\ 1 \\ l_1 \\ l_2 \end{array}\begin{array}{c} \begin{array}{ccc} 1 & l_2 & l_1 \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & \square & \text{◺} \\ 0 & \text{◹} & \square \end{array}\right] \end{array}, \tag{3.1}$$

where $l_1 = (m/n_s)(n_s + 1) - n_s$ and $l_2 = k + n_s$. If this structure is largely ignored, applying $U$ or $V$ amounts to a single GEMM with one of the factors being an $(l_1 + l_2) \times (l_1 + l_2)$ matrix. If, on the other hand, the triangular block structure is fully exploited, applying $U$ or $V$ amounts to two triangular matrix-matrix multiplies (TRMMs), one with an $l_1 \times l_1$ factor and the other with an $l_2 \times l_2$ factor, as well as two rectangular GEMMs, one with an $l_1 \times l_2$ factor and the other with an $l_2 \times l_1$ factor. The ratio between the flops needed by these two options is $1+(l_1^2+l_2^2)/(l_1^2+l_2^2+4l_1l_2)$. Following the suggestion in [7], we set the number of steps the bulge chain is chased to $k = 3/2m$, leading to $l_2 \approx 3/2l_1$. In this case, exploiting the triangular block structure reduces the number of flops by 26%. Whether this reduction leads to an actual saving of execution time depends on the performance of TRMM relative to GEMM, which may vary depending on BLAS implementations used for the target architecture and actual matrix sizes (e.g., see [23, 24]). A recent report [19] has identified computing environments for which TRMM performs significantly worse than GEMM, especially for the matrix dimensions arising in our application. In such a setting, it is more favorable to apply $U$ or $V$ with a single GEMM. However, many BLAS implementations, including the one proposed in [19], contain TRMM operations that perform well in comparison to GEMM. In this case, it is often possible to turn the flop reduction offered by the block triangular structure into an actual decrease of execution time.

As for the tiny-bulge multishift QR algorithm, we have to be aware of so-called [illegible] [7, 50], i.e., zero or tiny subdiagonal elements in $H$ that arise during the chasing process. In order to preserve the information contained in the bulge pairs, the chain of bulge pairs must be reintroduced in the row in which the zero appears. Fortunately, we do not have to be aware of zero or tiny subdiagonal elements in $T$, since the bulge pairs are properly passed through infinite eigenvalues; see section 4.4.

After a certain number of steps, the bottom bulge pair of the chain reaches the bottom right corners of the matrix pair. As soon as this happens, the whole chain is bulge-by-bulge chased off this corner, similarly to the introduction of bulge pairs.

**3.3. Classic deflation of finite eigenvalues.** The goal of (multishift) QZ iterations is to drive the subdiagonal entries of the Hessenberg matrix in $(H, T)$ to zero while preserving the upper triangular shape of $T$. Once a subdiagonal entry $h_{k+1,k}$ is considered zero, the problem is deflated into two smaller problems:

$$\left( \begin{bmatrix} H_{11} & H_{12} \\ 0 & H_{22} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \right).$$

Afterward, the (multishift) QZ iteration is applied separately to the $k \times k$ and $(n-k) \times (n-k)$ matrix pairs $(H_{11}, T_{11})$ and $(H_{22}, T_{22})$, respectively.

In the original formulation of the QZ algorithm [37] and the current implementation in LAPACK, a subdiagonal entry $h_{k+1,k}$ is considered zero if

$$|h_{k+1,k}| \le \mathbf{u}\, \|H\|_F. \tag{3.2}$$

A more conservative criterion, in the spirit of the LAPACK implementation of the QR algorithm, is to consider $h_{k+1,k}$ zero if

$$|h_{k+1,k}| \le \mathbf{u}\,(|h_{k,k}| + |h_{k+1,k+1}|). \tag{3.3}$$

It is known for standard eigenvalue problems that, especially in the presence of graded matrices, the use of the criterion (3.3) gives higher accuracy in the computed eigenvalues [41]. We have observed similar accuracy improvements for the QZ algorithm when using (3.3) in favor of (3.2). We have also encountered examples where both criteria give similar accuracy but with slightly shorter execution times for (3.2) due to earlier deflations.

**4. Dealing with infinite eigenvalues.** If the degree $p$ of the polynomial det $(\beta A - \alpha B)$ is less than $n$ then the matrix pair $(A, B)$ is said to have $n - p$ infinite eigenvalues. The relationship between infinite eigenvalues and the QZ algorithm is subtle and calls for caution. In finite-precision arithmetic, the QZ algorithm may utterly fail to correctly identify infinite eigenvalues, especially if the [illegible] of the matrix pair, defined as the size of the largest Jordan block associated with an infinite eigenvalue [17], is larger than one [37]. In the context of differential-algebraic equations (DAEs), the index of $(A, B)$ corresponds to the index of the DAE $B\dot{x} = Ax + f$. Many applications, such as multibody systems and electrical circuits, lead to DAEs with index at least two; see, e.g., [8, 39, 43].

If the matrix pair $(A, B)$ has an infinite eigenvalue then the matrix $B$ is singular. This implies that at least one of the diagonal entries in the upper triangular matrix $T$ in the Hessenberg-triangular form $(H, T)$ and in the generalized Schur form $(S, T)$ is zero, and vice versa. In finite-precision arithmetic, zero diagonal entries are spoiled by roundoff errors. While a tiny zero diagonal entry of $T$ implies that $T$ is numerically singular, the converse is generally [illegible] true. There are well-known examples of upper triangular matrices that are numerically singular but have diagonal entries that are not significantly smaller than the norm of the matrix [18, Ex. 5.5.1].

In such cases, much more reliable decisions on the nature of infinite eigenvalues can be met using algorithms that reveal Kronecker structures, such as GUPTRI [13, 14]. In some cases, infinite eigenvalues can be cheaply and reliably deflated by exploiting the structure of $A$ and $B$.

**4.1. Preprocessing deflation of infinite eigenvalues.** Given a regular matrix pair $(A, B)$ with infinite eigenvalues corresponding to several Jordan blocks, the QZ algorithm will typically compute eigenvalue pairs $(\alpha_i, \beta_i)$ with $\beta_i$ nonzero. Moreover, otherwise well-conditioned eigenvalues can be affected by perturbations from these defective infinite eigenvalues; e.g., they may coincide or appear in clusters of eigenvalues corresponding to computed infinite as well as finite eigenvalues. In the following, we briefly describe two preprocessing techniques for handling such situations.

**Exploiting staircase algorithms.** Without having any knowledge of the Jordan structure of the infinite eigenvalue, in principle, the only reliable and robust way to identify all infinite eigenvalues is to apply a preprocessing step with a staircase-type algorithm.

By applying the GUPTRI algorithm [13, 14, 27] to a regular pair $(A, B)$ with infinite eigenvalues, we get

$$U^T(A,B)V = \left( \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{\text{inf}} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{\text{inf}} \end{bmatrix} \right), \tag{4.1}$$

where $U$ and $V$ are orthogonal transformation matrices, $(A_{\text{inf}}, B_{\text{inf}})$ reveals the Jordan structure of the infinite eigenvalue, and $(A_{11}, B_{11})$ is a matrix pair with only finite eigenvalues.

Let us illustrate the GUPTRI form (4.1) with a small example. We consider a 7×7 pair $(A, B)$ with three finite eigenvalues and an infinite eigenvalue of multiplicity four corresponding to two nilpotent Jordan blocks $N_1$ and $N_3$. The infinite eigenvalue is both [illegible] and [illegible], since it has more than one eigenvector (two Jordan blocks) but lacks a full setting of eigenvectors (four Jordan blocks). Then $(A_{\text{inf}}, B_{\text{inf}})$ has the following schematic staircase form:

$$(A_{\text{inf}}, B_{\text{inf}}) = \left( \left[ \begin{array}{c|c|cc} \mathbf{z} & y & x & x \\ \hline 0 & \mathbf{y} & x & x \\ \hline 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} \end{array} \right], \left[ \begin{array}{c|c|cc} 0 & \mathbf{y} & x & x \\ \hline 0 & 0 & \mathbf{x} & \mathbf{x} \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right] \right).$$

The bold numbers $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ in $A_{\text{inf}}$ represent diagonal blocks of full rank, and the $\mathbf{x}$ and $\mathbf{y}$ in $B_{\text{inf}}$ represent superdiagonal blocks of full row rank. Outgoing from the bottom right corner of $B_{\text{inf}}$, the sizes of the diagonal blocks (stairs) $w = (2, 1, 1)$ are the [illegible] of the infinite eigenvalue. These indices relate to the dimensions of the nullspaces $\mathcal{N}(B^j)$ such that $\sum_{k=1}^{j} w_k = \dim \mathcal{N}(B^j)$ for $j = 1, 2, 3$. In other words, $w_j$ is the number of Jordan blocks of size $\geq j$. Now, the infinite Jordan structure can be read off from $w$ giving the [illegible] $s = (3, 1)$, where $s_1$ is the size of the largest Jordan block, $s_2$ is the size of the second largest block, and so on. Both $w$ and $s$ sum up to the [illegible] and $w_1$ is the [illegible] of the infinite eigenvalue.

After such a preprocessing deflation of the infinite eigenvalues of $(A, B)$, we apply the QZ algorithm to the matrix pair $(A_{11}, B_{11})$ in (4.1). For more introductory material on singular matrix pairs and the GUPTRI form see [27] and the references therein.

**Exploiting knowledge of structure.** In some cases, infinite eigenvalues can be reliably deflated by taking into account knowledge on the structure of the matrices $A$ and $B$. If this is feasible by orthogonal transformations, this is the recommended way of dealing with infinite eigenvalues, as the decision which eigenvalues are considered infinite is not affected by roundoff error. In the context of DAEs, several frameworks have been developed that can help identify and exploit such structures; see, e.g., [20, 35]. The following example is closely related to work by Stykel [42], in which $(A, B)$ arises from a semidiscretized Stokes equation.

[illegible] 4.1. Consider $A = \left[\begin{smallmatrix} K & L \\ L^T & 0 \end{smallmatrix}\right]$ and $B = \left[\begin{smallmatrix} M & 0 \\ 0 & 0 \end{smallmatrix}\right]$, where $L$ is an $m \times (n-m)$ matrix of full column rank ($n \le 2m$) and $M$ is an $m \times m$ symmetric positive definite matrix. By a QR decomposition of $L$ we may transform the matrix pair $(A, B)$ to

$$(A, B) \leftarrow \left( \begin{bmatrix} K_{11} & K_{12} & L_1 \\ K_{21} & K_{22} & 0 \\ L_1^T & 0 & 0 \end{bmatrix}, \begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \right),$$

where $L_1$ is an $m \times m$ invertible matrix. The submatrix $M_{22}$ is again symmetric positive definite, which in particular yields its invertibility. By a simple block permutation, $A$ and $B$ can be transformed to block upper triangular form,

$$(A, B) \leftarrow \left( \begin{bmatrix} L_1 & K_{12} & K_{11} \\ 0 & K_{22} & K_{21} \\ 0 & 0 & L_1^T \end{bmatrix}, \begin{bmatrix} 0 & M_{12} & M_{11} \\ 0 & M_{22} & M_{21} \\ 0 & 0 & 0 \end{bmatrix} \right).$$

Thus, the eigenvalues of the matrix pair $(K_{22}, M_{22})$ constitute the finite eigenvalues of $(A, B)$.

**4.2. Deflation of infinite eigenvalues within the QZ algorithm.** Although preprocessing is the preferable way of dealing with infinite eigenvalues, there can be good reasons to let the QZ algorithm do this job, particularly if the reliable detection of infinite eigenvalues is not a major concern. One reason is that computing a GUPTRI form is quite a costly procedure [14]. This and the following two subsections are concerned with existing and new approaches to deflate infinite eigenvalues that are signaled by tiny diagonal entries of the matrix $T$ in a Hessenberg-triangular matrix pair $(H, T)$.

For testing the smallness of a diagonal entry $t_{jj}$ we may, similar to (3.2)–(3.3), either use the norm-wise criterion

$$|t_{jj}| \le \mathbf{u} \cdot \|T\|_F, \tag{4.2}$$

as implemented in the LAPACK routine `DHGEQZ`, or the neighbor-wise criterion $|t_{jj}| \le \mathbf{u} \cdot (|t_{j-1,j}| + |t_{j,j+1}|)$. The latter criterion might help avoid artificial infinite eigenvalues caused by a poor scaling of the matrix pair. Let us briefly sketch the procedure developed by Moler and Stewart [37] for deflating an infinite eigenvalue after $t_{jj}$ has been set to zero, for the case $n = 5$ and $j = 3$:

$$(H, T) = \left( \begin{bmatrix} h & h & h & h & h \\ h & h & h & h & h \\ 0 & h & h & h & h \\ 0 & 0 & h & h & h \\ 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} t & t & t & t & t \\ 0 & t & t & t & t \\ 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & t \end{bmatrix} \right).$$

First, a Givens rotation is applied to columns 2 and 3 to annihilate $t_{22}$, followed by a Givens rotation acting on rows 3 and 4 to annihilate the newly introduced nonzero entry $h_{42}$:

$$(H, T) \leftarrow \left( \begin{bmatrix} h & \hat{h} & \hat{h} & h & h \\ h & \hat{h} & \hat{h} & h & h \\ 0 & \hat{h} & \hat{h} & \hat{h} & \hat{h} \\ 0 & \hat{0} & \hat{h} & \hat{h} & \hat{h} \\ 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} t & \hat{t} & \hat{t} & t & t \\ 0 & \hat{0} & \hat{t} & t & t \\ 0 & 0 & 0 & \hat{t} & \hat{t} \\ 0 & 0 & 0 & \hat{t} & \hat{t} \\ 0 & 0 & 0 & 0 & t \end{bmatrix} \right).$$

In a similar manner, the two zero diagonal entries in $T$ are pushed one step upward:

$$(H,T) \leftarrow \left( \begin{bmatrix} \hat{h} & \hat{h} & h & h & h \\ \hat{h} & \hat{h} & \hat{h} & \hat{h} & \hat{h} \\ \hat{0} & \hat{h} & \hat{h} & \hat{h} & \hat{h} \\ 0 & 0 & h & h & h \\ 0 & 0 & 0 & h & h \end{bmatrix}, \begin{bmatrix} \hat{0} & \hat{t} & t & t & t \\ 0 & 0 & \hat{t} & \hat{t} & \hat{t} \\ 0 & 0 & \hat{t} & \hat{t} & \hat{t} \\ 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & t \end{bmatrix} \right).$$

Finally, a Givens rotation acting on rows 1 and 2 is used to deflate the infinite eigenvalue at the top left corner:

$$(H,T) \leftarrow \left( \left[ \begin{array}{c|cccc} \hat{h} & \hat{h} & \hat{h} & \hat{h} & \hat{h} \\ \hline \hat{0} & \hat{h} & \hat{h} & \hat{h} & \hat{h} \\ 0 & h & h & h & h \\ 0 & 0 & h & h & h \\ 0 & 0 & 0 & h & h \end{array} \right], \left[ \begin{array}{c|cccc} 0 & \hat{t} & \hat{t} & \hat{t} & \hat{t} \\ \hline 0 & \hat{t} & \hat{t} & \hat{t} & \hat{t} \\ 0 & 0 & t & t & t \\ 0 & 0 & 0 & t & t \\ 0 & 0 & 0 & 0 & t \end{array} \right] \right).$$

The outlined procedure requires roughly $6jn$ flops for updating each of the factors $H, T, Q$, and $Z$. If $j > n/2$, it is cheaper to push the infinite eigenvalue to the bottom right corner.

**4.3. Windowing techniques for deflating infinite eigenvalues.** The algorithm described in the previous subsection performs $\mathcal{O}(jn)$ flops while accessing $\mathcal{O}(jn)$ memory, making it costly in terms of execution time on computing systems with deep memory hierarchies. If the dimension of the matrix pair is large and many infinite eigenvalues are to be deflated, this degrades the overall performance of the multishift QZ algorithm. A higher computation/communication ratio can be attained by using windowing techniques similar to those proposed in [5, 12, 32]. In the following, we illustrate such an algorithm, conceptually close to a recently presented block algorithm for reordering standard and generalized Schur forms [32].

Consider a matrix pair $(H,T)$ in Hessenberg-triangular form, where the 9th and the 16th diagonal entries of $T$ are zero; see Figure 4.1(a). Both zero entries will be pushed simultaneously in a window-by-window fashion to the top left corner. The first step consists of pushing the lower zero diagonal entry to the top left corner of the 8-by-8 window marked by the light gray area in Figure 4.1(b). This creates zero diagonal entries at positions 11 and 12. Note that it makes little sense to push one step further; the leading zero at position 10 would be destroyed when pushing the zero diagonal entry at position 9. During this procedure, only the entries of $H$ and $T$ that reside within the window are updated and the data representing the performed Givens rotations is pipelined; see [32] for more details. Afterward, the pipelined transformations are applied to the parts outside the window marked by dark gray areas in Figure 4.1(b) as well as to the corresponding parts of the transformation matrices $Q$ and $Z$. To maintain locality of the memory reference pattern, rows are updated in stripes of $n_b$ columns (in the computational environments we considered, choosing $n_b = 32$ was nearly optimal). The next window contains the diagonal positions $5, \ldots, 12$; see Figure 4.1(c). The zeros at positions 9 and 11 are subsequently pushed to positions 5 and 7, respectively. Again, the update of parts outside the window as well as the update of the transformation matrices are delayed as described above. The last 8-by-8 window resides in the top left corner and yields the deflation of two infinite eigenvalues; see Figure 4.1(d).

Note that we have only provided the generic picture; pushing a zero diagonal entry in $T$ may leave "traces" in the form of additional zero diagonal entries. A proper implementation of the windowing algorithm has to take care of such events.

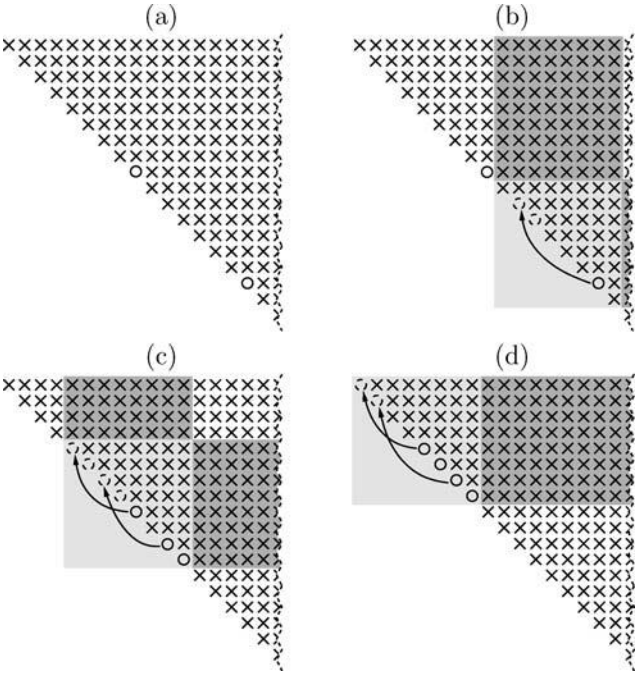


FIG. 4.1. *Illustration of windowing algorithm for deflating two infinite eigenvalues. Only the $T$-matrix of $(H,T)$ is shown.*

Moreover, to achieve optimal performance, the number of zero diagonal entries to be simultaneously pushed and the window size should be significantly larger than those chosen in the descriptive example; see section 7.2.

**4.4. Infinite eigenvalues and multishift QZ iterations.** An important observation made in [53] is that the shift transmission mechanism works even if $T$ is singular, provided that this singularity does not affect the generalized Francis shifts or the definition of the first column of the shift polynomial. In fact, Theorem 2.4 assumes only that the intended shifts are finite and that the $m$th leading principal submatrix of $T$ is nonsingular.

Hence, zero diagonal entries at positions $m+1,\dots,n-m$ in $T$ do not affect the information contained in the bulge pairs and consequently do not affect the convergence of the QZ iteration. What happens to such a zero diagonal entry if a bulge pair passes through it? This question has been addressed by Ward [46, 47] for $m \leq 2$, and by Watkins for general $m$ [53]. The answer is that the zero diagonal entry moves $m$ positions upward along the diagonal. Note that although it is assumed in [53] that the multishift QZ iteration is based on Givens rotations, the same answer holds for a QZ iteration based on (opposite) Householder matrices; see [31].

These results imply that infinite eigenvalues need only be deflated if they correspond to zero diagonal entries at positions $1,\dots,m$ and $n-m+1,\dots,n$ of $T$. Other zero diagonal entries will be automatically moved upward in the course of the QZ algorithm to the top diagonal positions, where they then can be deflated. Note, however, that this "transport" of zero diagonal elements holds only under the assumption of exact arithmetic; it can be severely affected by roundoff error.

[illegible] 4.2. Consider the $10 \times 10$ matrix pair

$$(H,T) = \left( \begin{bmatrix} 3 & 3 & \cdots & \cdots & 3 \\ 1 & 3 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & 3 & 3 \\ & & & 1 & 3 \end{bmatrix}, \begin{bmatrix} 1 & 1 & \cdots & \cdots & 1 \\ & 0 & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & 0 & 1 \\ & & & & 1 \end{bmatrix} \right)$$

in Hessenberg-triangular form. It can be shown that this matrix pair has four infinite eigenvalues. Applying a single-shift QZ iteration, Algorithm 1 with $m = 1$, once to $(H,T)$ leads to an updated triangular matrix $T$ with the leading diagonal entry being exactly zero. None of the other diagonal entries, however, satisfies the deflation criterion (4.2). Also in the course of further QZ iterations applied to the deflated matrix pair no other infinite eigenvalue can be detected. After convergence, the three remaining infinite eigenvalues of $(H,T)$ have been perturbed to finite eigenvalues of magnitude $\approx 1.9 \times 10^5$. On the other hand, if all entries of $T$ satisfying (4.2) are subsequently deflated [illegible] any QZ iteration, then all four infinite eigenvalues are detected.

Example 4.2 reveals that not taking care of all (nearly) zero diagonal entries in $T$ increases the chances that infinite eigenvalues go undetected. Besides the obvious disadvantages, failing to detect infinite eigenvalues may have an adverse effect on the convergence of the QZ algorithm [29, 53]. There is no simple cure for the effects observed in Example 4.2. Setting a diagonal entry, which is known to be zero in exact arithmetic but does not satisfy (4.2), explicitly to zero would spoil the backward stability of the QZ algorithm. We therefore recommend taking care of [illegible] nearly zero diagonal entries in $T$ before applying a QZ iteration. Small or—in rare circumstances—even zero diagonal entries in $T$ may still appear during a multishift QZ iteration. In particular, we may encounter such a situation when having chased some but not all of the bulge pairs from a chain of bulge pairs. Then the small diagonal entry resides between two smaller chains and from the point of view of Example 4.2 it would be desirable to deflate the corresponding (nearly) infinite eigenvalue. However, with the existing deflation techniques, this can only be achieved by chasing off at least one of the smaller chains.

**5. Singular and nearly singular pencils.** For a moment, let us consider a square singular pencil $\beta A - \alpha B$. Then the generalized Schur form $(S,T)$ of $(A,B)$ must (in theory) have at least one pair $(s_{ii}, t_{ii})$ with $s_{ii} = t_{ii} = 0$. This situation appears, for example, when $A$ and $B$ have a common column (or row) null space. On the other hand, given a singular pair $(S,T)$ in generalized Schur form with a regular part, an equivalence transformation of $(S,T)$ that produces upper triangular matrices may give no information about the regular part by inspection of the diagonal elements. For example, the pair

$$(S,T) = \left( \left[\begin{array}{cc|c|c} 3 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ \hline 0 & 0 & 2 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right], \left[\begin{array}{cc|c|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \right)$$

has the finite eigenvalues 3/1, 3/1, and 2/1, besides the singular part (0/0). The equivalent matrix pair

$$(SQ,TQ) = \left( \left[\begin{array}{c|cc|c} 0 & 3 & 1 & 0 \\ 0 & 0 & 3 & 0 \\ \hline 0 & 0 & 0 & 2 \\ \hline 0 & 0 & 0 & 0 \end{array}\right], \left[\begin{array}{c|cc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 \end{array}\right] \right), \text{ with } Q = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

has all diagonal elements equal to zero (0/0). So examination of the diagonal elements only gives no indication of the well-defined regular part of $(S, T)$.

In practice, the QZ algorithm will in general not detect the above-mentioned singularities reliably and otherwise well-conditioned eigenvalues can change drastically, meaning that the values of the computed pairs $(s_{ii}, t_{ii})$ cannot be trusted (e.g., see Wilkinson [54] for several illustrative examples.) Moreover, it is impossible to decide just by inspection whether $s_{ii} = \epsilon_1$ and $t_{ii} = \epsilon_2$, with $\epsilon_1$ and $\epsilon_2$ tiny, correspond to a finite eigenvalue $\epsilon_1/\epsilon_2$ or to a true singular pencil. Anyhow, with this information we know that $\beta A - \alpha B$ is close to a singular pencil. (Note that the converse of this statement is not true [9, 27].)

Although the QZ algorithm delivers erratic results for singular or almost singular cases, the computed results are still exact for small perturbations of the original matrix pair $(A, B)$. To robustly deal with such cases, it is recommended to first identify any singularity and deflate the associated Kronecker structure of $(A, B)$ in a preprocessing step before the QZ algorithm is applied. As with infinite eigenvalues, this can be done by exploiting staircase-type algorithms like GUPTRI [13, 14].

## 6. Aggressive early deflation applied to the QZ algorithm.

The idea behind the aggressive early deflation strategy in the QZ algorithm is to enhance the deflation strategy described in section 3.3 by taking advantage of perturbations outside the subdiagonal entries of the Hessenberg matrix, as in the QR algorithm [6]. This gives the possibility to identify and deflate converged eigenvalues much earlier than either of the deflation criteria (3.2) and (3.3) would do, which results in fewer QZ iterations and thereby has the potential to save both floating point operations and execution time.

### 6.1. Pairs of reducing perturbations.

For simplicity, we consider an $n \times n$ [illegible] unreduced Hessenberg-triangular matrix pair $(H, T)$. Let $P_H$ and $P_T$ be complex perturbation matrices. Suppose there exist a unitary matrix $Q$ of the form $Q = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q} \end{bmatrix}$ and a unitary matrix $Z$ such that the transformed perturbed matrix pair,

$$(\hat{H}, \hat{T}) \equiv Q^H (H + P_H, T + P_T) Z, \tag{6.1}$$

is in reduced Hessenberg-triangular form:

$$\hat{H} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ 0 & \hat{H}_{22} \end{bmatrix}, \qquad \hat{T} = \begin{bmatrix} \hat{T}_{11} & \hat{T}_{12} \\ 0 & \hat{T}_{22} \end{bmatrix}. \tag{6.2}$$

Then, in analogy to the matrix case, $(P_H, P_T)$ is called a [illegible]. If the norm of $(P_H, P_T)$ is tiny, the equivalence transformation above has split the problem of computing the eigenvalues of $(H, T)$ in two (or more) subproblems of smaller size without affecting the backward stability of the QZ algorithm.

In the following, we derive results that characterize and identify pairs of reducing perturbations, which are extensions of similar results for the matrix case [6].

LEMMA 6.1. [illegible] $(H, T)$ [illegible] $H, T \in \mathbb{C}^{n \times n}$ [illegible] $P_H, P_T \in \mathbb{C}^{n \times n}$ [illegible] $T + P_T$ [illegible] $(P_H, P_T)$ [illegible] reducing perturbation pair [illegible] $(H, T)$ if and only if [illegible] $(H + P_H, T + P_T)$ [illegible] $y \in \mathbb{C}^n$ [illegible] $y_1 = 0$

[illegible] Assume $(P_H, P_T)$ is a reducing perturbation pair for $(H, T)$; i.e., there exist a unitary matrix $Q = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q} \end{bmatrix}$ and a unitary matrix $Z$ such that $(\hat{H}, \hat{T})$ defined by (6.1) is in reduced block-triangular form (6.2). This implies that $(\hat{H}, \hat{T})$ has a left (generalized) eigenvector $\hat{y}$ with $\hat{y}_1 = 0$; indeed, the first $\dim(\hat{H}_{11})$ components equal to zero. Since $Q$ has block diagonal structure, it follows that $y = Q\hat{y}$ is a left eigenvector of $(H + P_H, T + P_T)$ with $y_1 = \hat{y}_1 = 0$.

In the opposite direction, assume that $(H + P_H, T + P_T)$ has a left (generalized) eigenvector $y \in \mathbb{C}^n$ with a zero first component, $y_1 = 0$, associated with the eigenvalue pair $(\alpha, \beta) \in \mathbb{C}^2$, i.e., $\beta y^H(H + P_H) = \alpha y^H(T + P_T)$. By replacing the initial QR factorization of $B$ in the standard algorithm for reducing a matrix pair $(A, B)$ to Hessenberg-triangular form [18, Alg. 7.7.1] by an RQ factorization of $B$, we construct unitary matrices $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & Q \end{smallmatrix}\right]$ and $Z$ such that $Q^H(H + P_H, T + P_T)Z = (\hat{H}, \hat{T})$ is in Hessenberg-triangular form. It follows that $\hat{y} = Q^H y$ is a left (generalized) eigenvector of $(\hat{H}, \hat{T})$ and $\hat{y}_1 = y_1 = 0$ due to the fact that $Q$ is block diagonal. Let $k$ be the smallest index for which $\hat{y}_k \neq 0$ and partition $\hat{y}^H = [0, z]^H$ with $z \in \mathbb{C}^{n-k+1}$. If $\hat{H}$ and $\hat{T}$ are conformably partitioned,

$$\hat{H} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ h_{k,k-1} e_1 e_{k-1}^T & \hat{H}_{22} \end{bmatrix}, \qquad \hat{T} = \begin{bmatrix} \hat{T}_{11} & \hat{T}_{12} \\ 0 & \hat{T}_{22} \end{bmatrix},$$

then $\beta\, \hat{y}^H \hat{H} = \alpha\, \hat{y}^H \hat{T}$ yields

$$\beta\, [\hat{h}_{k,k-1} \hat{y}_k e_{k-1}^T, z^H \hat{H}_{22}] = \alpha\, [0, z^H \hat{T}_{22}].$$

The nonsingularity of $T + P_T$ implies $\beta \neq 0$, which in turn gives $\hat{h}_{k,k-1} = 0$; i.e., $\hat{H}$ is in reduced Hessenberg form. □

Note that the second part of the proof of Lemma 6.1 also shows how orthogonal matrices $Q$ and $Z$ yielding a deflated matrix pair (6.2) can be obtained by a slightly modified form of Hessenberg-triangular reduction. In the context of aggressive deflation, a useful reducing perturbation pair $(P_H, P_T)$ must have enough zero structure so that relatively little work is needed to retransform $(H + P_H, T + P_T)$ to Hessenberg-triangular form. By restricting $P_H$ and $P_T$ to Hessenberg and triangular matrices, respectively, there will be no extra work.

LEMMA 6.2. $(P_H, P_T)$ [illegible] reducing perturbation pair [illegible] $(H, T)$ [illegible] if and only if $P_T$ [illegible] $P_H$ [illegible]

[illegible] Let $(P_H, P_T)$ be a reducing perturbation pair in Hessenberg-triangular form. Decompose $P_H = P_H^{(s)} + P_H^{(u)}$ in its subdiagonal part $P_H^{(s)}$ and its upper triangular part $P_H^{(u)}$. Then $(P_H^{(s)}, 0)$ is a reducing perturbation pair of smaller norm. □

This choice leads to the small-subdiagonal deflation strategy for the QZ algorithm described in section 3.3.

In order to reach a more aggressive deflation strategy, we must allow more general perturbations, where $(P_H, P_T)$ is not necessarily in Hessenberg-triangular form. Extending the matrix case, we consider small perturbations $P_H$ and $P_T$ that are nonzero only in the last $k$ rows and $k + 1$ columns. Now, if $k \ll n$, the cost is small (compared to a QZ iteration) to retransform the perturbed matrix pair to Hessenberg-triangular form; see also section 6.2.

Let the matrix pair $(H,T)$ be in unreduced Hessenberg-triangular form with $H,T \in \mathbb{C}^{n\times n}$ and partitioned as follows:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix}, \qquad T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ 0 & T_{22} & T_{23} \\ 0 & 0 & T_{33} \end{bmatrix}, \tag{6.3}$$

where the block rows from top to bottom (and block columns from left to right) have $n-n_w-1, 1$, and $n_w$ rows (columns), respectively. Let the perturbation pair $(P_H,P_T)$ be partitioned conformably with $(H,T)$, but with the following nonzero structure:

$$P_H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & P_{32}^{(H)} & P_{33}^{(H)} \end{bmatrix}, \qquad P_T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & P_{33}^{(T)} \end{bmatrix}. \tag{6.4}$$

A special choice of such perturbations is given in the following lemma.

LEMMA 6.3. [illegible] $(\alpha,\beta)$ [illegible] $(H_{33},T_{33})$ [illegible] $y$ [illegible] $\|y\|_2 = 1$ [illegible] $(P_H,P_T)$ [illegible] (6.4) [illegible] $P_{32}^{(H)} = -(y^H H_{32})y$ [illegible] $P_{33}^{(H)} = 0$ [illegible] $P_T = 0$ [illegible].

[illegible] We have

$$\beta\,[0,0,y^H](H+P_H) = \beta\,[0,0,y^H H_{33}] = \alpha\,[0,0,y^H T_{33}] = \alpha\,[0,0,y^H]T.$$

This shows that $[0,0,y^H]^H$ is a left eigenvector of the perturbed matrix pair $(H+P_H, T+P_T)$ with $P_T = 0$, which together with Lemma 6.1 concludes the proof. ☐

To search for reducing perturbation pairs, we can choose from all, generically $n_w$, possible perturbations in the sense of Lemma 6.3. Although this strategy will generally yield only a reducing perturbation of [illegible] minimal Frobenius norm among all pairs of the form (6.4), the perturbations of Lemma 6.3 have the major advantage of being effectively computed and tested. Finding the minimum among all reducing perturbations of the form (6.4) is closely related to finding the distance to uncontrollability of a descriptor system [10]. This connection along with numerical methods for computing the distance to uncontrollability will be studied in a forthcoming paper. However, in preliminary numerical experiments with the multishift QR algorithm we observed that rarely can any extra deflations be gained from using perturbations more general than those of Lemma 6.3.

To illustrate the effectiveness of Lemma 6.3, let us consider the following matrix pair, which has been considered in [1] as an extension of the motivating example in [6]:

$$(H,T) = \left( \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 0.001 & 1 & 0 & 0 & 0 & 0 \\ & 0.001 & 2 & 0 & 0 & 0 \\ & & 0.001 & 3 & 0 & 0 \\ & & & 0.001 & 4 & 0 \\ & & & & 0.001 & 5 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & 1 & 0 & 0 \\ & & & & 1 & 0 \\ & & & & & 1 \end{bmatrix} \right). \tag{6.5}$$

Let us consider a partitioning of the form (6.3) for $n_w = 5$. Then the eigenvalues of $(H_{33},T_{33})$ are given by the $(\alpha,\beta)$ pairs $(1,1),(2,1),\ldots,(5,1)$ with $\lambda = \alpha/\beta$. Each of these eigenvalues yields a reducing perturbation pair in the sense of Lemma 6.3. The respective norms of $\|P_{32}^{(H)}\|$ are as follows:

$$1:\ 1.0\times 10^{-3},\ 2:\ 1.0\times 10^{-6},\ 3:\ 5.0\times 10^{-10},\ 4: 1.7\times 10^{-13},\ 5: 4.2\times 10^{-17}.$$

In double precision, the eigenvalue 5 can thus be safely deflated. In single precision, even three eigenvalues (3, 4, and 5) correspond to a reducing perturbation of norm below machine precision. See also section 7.3, where this example is studied for larger matrices.

**6.2. Implementation aspects.** In the following, we describe an efficient method which puts the aggressive early deflation motivated by Lemma 6.3 into practice. For this purpose, we consider a partition of $(H, T)$ of the form (6.3) and focus on the $n_w \times (n_w + 1)$ submatrix pair $([H_{32},\ H_{33}], [0,\ T_{33}])$, which defines the [illegible]

First, by means of the QZ algorithm, orthogonal matrices $Q_1$ and $Z_1$ resulting in a generalized Schur decomposition of $(H_{33}, T_{33})$ are computed. This admits the following partitioning:

$$Q_1^T([H_{32},\ H_{33}], [0,\ T_{33}]) \begin{bmatrix} 1 & 0 \\ 0 & Z_1 \end{bmatrix} = \left( \begin{bmatrix} s_3 & \tilde{H}_{33} & \tilde{H}_{34} \\ s_4 & 0 & \tilde{H}_{44} \end{bmatrix}, \begin{bmatrix} 0 & \tilde{T}_{33} & \tilde{T}_{34} \\ 0 & 0 & \tilde{T}_{44} \end{bmatrix} \right),$$

where $s_3, s_4$ are column vectors of appropriate size, and $(\tilde{H}_{44}, \tilde{T}_{44})$ is either $1 \times 1$, representing a real eigenvalue of $(H_{33}, T_{33})$, or $2 \times 2$, representing a complex conjugate pair of eigenvalues. If $(\tilde{H}_{44}, \tilde{T}_{44})$ represents a real eigenvalue, then the corresponding reducing perturbation in the sense of Lemma 6.3 is obtained by setting the scalar $s_4$ to zero. Similarly, if $(\tilde{H}_{44}, \tilde{T}_{44})$ is $2 \times 2$, a reducing perturbation is obtained by setting the two entries of $s_4$ to zero. This is, strictly speaking, not a perturbation in the sense of Lemma 6.3 and it may happen that one of the two complex conjugate eigenvalues of $(\tilde{H}_{44}, \tilde{T}_{44})$ considered individually yields a reducing perturbation which is significantly smaller than $\|s_4\|_2$. However, deflating this eigenvalue alone is not possible without leaving the realm of real matrices.

There are several possible choices for criteria under which $\|s_4\|_2$, the norm of the reducing perturbation described above, can be considered negligible. A liberal deflation criterion, which just preserves numerical backward stability, is given by

$$\|s_4\|_2 \leq \mathbf{u}\|H\|_F. \tag{6.6}$$

A more conservative criterion in the spirit of (3.3) is given by

$$\|s_4\|_2 \leq \begin{cases} \mathbf{u}|\tilde{H}_{44}| & \text{if } \tilde{H}_{44} \text{ is } 1 \times 1, \\ \mathbf{u}\sqrt{|\det(\tilde{H}_{44})|} & \text{otherwise.} \end{cases} \tag{6.7}$$

This is preferred for reasons explained in section 3.3. A range of other criteria can be found in [6, sec. 2.4].

If $\|s_4\|_2$ satisfies the chosen deflation criterion, we mark $(\tilde{H}_{44}, \tilde{T}_{44})$ as deflatable and apply the described process again to the reduced matrix pair $([s_3,\ \tilde{H}_{33}], \tilde{T}_{33}])$. Otherwise, we mark $(\tilde{H}_{44}, \tilde{T}_{44})$ as undeflatable and reorder the generalized Schur decomposition of $(H_{33}, T_{33})$ to construct orthogonal matrices $Q_2$ and $Z_2$ such that

$$(Q_1 Q_2)^T([H_{32},\ H_{33}], [0,\ T_{33}]) \begin{bmatrix} 1 & 0 \\ 0 & Z_1 Z_2 \end{bmatrix} = \left( \begin{bmatrix} \bar{s}_3 & \bar{H}_{33} & \bar{H}_{34} \\ \bar{s}_4 & 0 & \bar{H}_{44} \end{bmatrix}, \begin{bmatrix} 0 & \bar{T}_{33} & \bar{T}_{34} \\ 0 & 0 & \bar{T}_{44} \end{bmatrix} \right),$$

where $(\bar{H}_{33}, \bar{T}_{33})$ is of the same order and has the same eigenvalues as $(\tilde{H}_{44}, \tilde{T}_{44})$. In this case, the described process is applied again to the matrix pair $([\bar{s}_4,\ \bar{H}_{44}], \bar{T}_{44}])$. The whole procedure is repeated until the matrix pair vanishes, i.e., $n_w - k$ undeflatable and $k$ deflatable eigenvalues have been found yielding a decomposition of the form

$$Q^T([H_{32},\ H_{33}], [0,\ T_{33}]) \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} = \left( \begin{bmatrix} \check{s}_3 & \check{H}_{33} & \check{H}_{34} \\ \check{s}_4 & 0 & \check{H}_{44} \end{bmatrix}, \begin{bmatrix} 0 & \check{T}_{33} & \check{T}_{34} \\ 0 & 0 & \check{T}_{44} \end{bmatrix} \right),$$

where $(\check{H}_{44}, \check{T}_{44})$ is $k \times k$ and contains all deflatable eigenvalues. Moreover, we have $\|\check{s}_4\|_2 \leq \sqrt{k}\mathbf{u}\|H\|_F$ no matter whether (6.6) or (6.7) is used. Hence, $\check{s}_4$ can be safely set to zero and the QZ algorithm is continued with the $(n-k) \times (n-k)$ matrix pair

$$(\tilde{H}, \tilde{T}) = \left( \begin{bmatrix} H_{11} & H_{12} & H_{13}Z \\ H_{21} & H_{22} & H_{23}Z \\ 0 & \check{s}_3 & \check{H}_{33} \end{bmatrix}, \begin{bmatrix} T_{11} & T_{12} & T_{13}Z \\ 0 & T_{22} & T_{23}Z \\ 0 & 0 & \check{T}_{33} \end{bmatrix} \right).$$

Note that the matrix pair $(\tilde{H}, \tilde{T})$ is not in Hessenberg-triangular form due to the "spike" $\check{s}_3$. If we apply a Householder matrix $\mathcal{H}_1(\check{s}_3) = I - \beta vv^T$ to the last $n_w - k$ rows of $(\tilde{H}, \tilde{T})$, we have

$$\mathcal{H}_1(\check{s}_3)\check{T}_{33} = \check{T}_{33} - \beta v(\check{T}_{33}^T v)^T.$$

Hence, $\mathcal{H}_1(\check{s}_3)\check{T}_{33}$ is a rank-one update of a triangular matrix. Similar to updating algorithms for the QR decomposition [18, sec. 12.5], we can construct an orthogonal matrix $Z_3$ as a sequence of $n_w - k - 1$ Givens rotations such that $Z_3^T \check{T}_{33}^T v = \gamma e_n$ for some $\gamma \in \mathbb{R}$. Consequently,

$$\mathcal{H}_1(\check{s}_3)\check{T}_{33}Z_3 = \check{T}_{33}Z_3 - \beta\gamma v e_n^T$$

is an upper Hessenberg matrix. By another sequence of $n_w - k - 1$ Givens rotations the subdiagonal elements of $\mathcal{H}_1(\check{s}_3)\check{T}_{33}Z_3$ can be eliminated so that $\mathcal{H}_1(\check{s}_3)\check{T}_{33}Z_3Z_4$ becomes upper triangular. The described algorithm requires $\mathcal{O}((n_w - k)^2)$ flops which is favorable compared to the $\mathcal{O}((n_w - k)^3)$ flops needed for computing an RQ factorization of $\mathcal{H}_1(\check{s}_3)\check{T}_{33}$ from scratch. Finally, the standard reduction algorithm [18, Alg. 7.7.1] without the initial QR factorization is applied to the matrix pair $\mathcal{H}_1(\check{s}_3)(\check{H}_{33}, \check{T}_{33})Z_3Z_4$ in order to compute orthogonal matrices $Q_3 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \tilde{Q}_3 \end{smallmatrix}\right]$ and $Z_5$ such that $Q_3^T \mathcal{H}_1(\check{s}_3)(\check{H}_{33}, \check{T}_{33})Z_3Z_4Z_5$ is Hessenberg-triangular. Setting

$$\tilde{Q} = \begin{bmatrix} I_{n-n_w} & 0 \\ 0 & \mathcal{H}_1(\check{s}_3)Q_3 \end{bmatrix}, \quad \tilde{Z} = \begin{bmatrix} I_{n-n_w} & 0 \\ 0 & Z_3Z_4Z_5 \end{bmatrix}$$

yields a Hessenberg-triangular matrix pair $\tilde{Q}^T(\tilde{H}, \tilde{T})\tilde{Z}$ from which the multishift QZ algorithm can be continued. Note that before continuing with a multishift QZ iteration it can be beneficial to apply aggressive early deflation again if sufficiently many eigenvalues have been deflated, i.e., if the ratio $k/n_w$ is above a certain threshold, which has been set to 40% in our experiments (parameter #3 in Table 7.1).

## 7. Computational experiments.

To assess their performance and robustness, we have implemented the newly developed variants of the QZ algorithm in FORTRAN 77 and performed several experiments to be described in the following subsections.

### 7.1. Computational platform(s).

The experiments are carried out on one processor of two of the HPC2N clusters, `seth` and `sarek`, which have advanced memory systems with different characteristics.

The cluster `seth` consists of 120 nodes, dual Athlon MP2000+ (1.667Ghz) with 1 GB memory per node. Athlon MP2000+ has a 64 kB instruction, a 64 kB data L1 Cache, and 256 kB of integrated L2 cache. Software used: Debian GNU/Linux 3.0, Portland F90 6.0, ATLAS BLAS 3.5.9.

The cluster `sarek` consists of 190 HP DL145 nodes, with dual AMD Opteron 248 (2.2GHz) and 8 GB memory per node. AMD Opteron 248 has a 64 kB instruction

Table 7.1
*Default values for some parameters of the multishift QZ algorithm with aggressive early deflation.*

| | | `seth` | `sarek` |
|---|---|---|---|
| #1 | Minimal (sub)matrix pair dimension for multishift QZ iterations | 300 | 300 |
| #2 | Minimal (sub)matrix pair dimension for aggressive early deflation | 300 | 300 |
| #3 | Minimal success rate for repeated aggressive early deflation | 40% | 40% |
| #4 | Window size for simultaneous deflation of infinite eigenvalues | 60 | 84 |
| #5 | Number of infinite eigenvalues to be deflated simultaneously | 20 | 28 |

and 64 kB data L1 Cache (2-way associative) and a 1024 kB unified L2 Cache (16-way associative). Software used: Debian GNU/Linux 3.1, Portland F90 6.0, Goto BLAS 0.94.

All results reported are run in double precision real arithmetic ($\epsilon_{mach} \approx 2.2 \times 10^{-16}$).

**7.2. Random matrix pairs.** The described multishift QZ iterations and deflation algorithms depend on various parameters, which all have some influence on the overall execution time of the QZ algorithm. We have performed numerical experiments with randomly generated matrix pairs and numerous sets of parameters to measure the influence of each individual parameter. In the following, we focus on the three parameters that have been observed to have the largest impact on the execution time and therefore require particular attention:

$m$: number of simultaneous shifts used in each multishift QZ iteration (integer multiple of $n_s$),
$n_s$: number of shifts contained in each bulge during multishift QZ iterations,
$n_w$: aggressive early deflation window size.

All other parameters turned out to have less influence on the performance and have been set in a heuristic manner. The default values displayed in Table 7.1 yielded good performance for matrix pairs of size $500, \dots, 3000$. If the order of an active submatrix pair in the course of the multishift QZ algorithm described in section 3 becomes smaller than parameter #1, it is more efficient to resort to double-shift QZ iterations. Similarly, if the order is smaller than parameter #2, aggressive early deflation is turned off. It is best to choose #1 not smaller than #2. If aggressive early deflation yielded the deflation of $k$ eigenvalues and the ratio $k/n_w$ exceeds parameter #3, another search for early deflations is immediately performed on the deflated matrix pair before applying a (multishift) QZ iteration. Finally, the parameters #4 and #5 represent the window size and the maximal number of infinite eigenvalues to be pushed simultaneously in the block algorithm for deflating infinite eigenvalues described in section 4.3. It is necessary to choose #4 larger than two times #5; we found choosing #4 three times larger nearly optimal.

To make the new implementation better comparable to the LAPACK version 3.0 implementation, we used throughout all experiments the liberal deflation criteria (3.2), (4.2), and (6.6). The use of the more conservative deflation criteria (3.3) and (6.7) may result in more accurate eigenvalues but may also lead to slightly more QZ iterations.

**7.2.1. Influence of $\boldsymbol{m}$ and $\boldsymbol{n_s}$.** To measure the influence of the parameters $m$ and $n_s$ on the performance of the multishift QZ algorithm [illegible] aggressive early deflation, we generated $n \times n$ matrices $A$ and $B$ having pseudorandom entries uniformly distributed in the interval $[-1, 1]$ and reduced them to Hessenberg-triangular form by applying the LAPACK version 3.0 routine `DGGHRD`. Figures 7.1 and 7.2 display

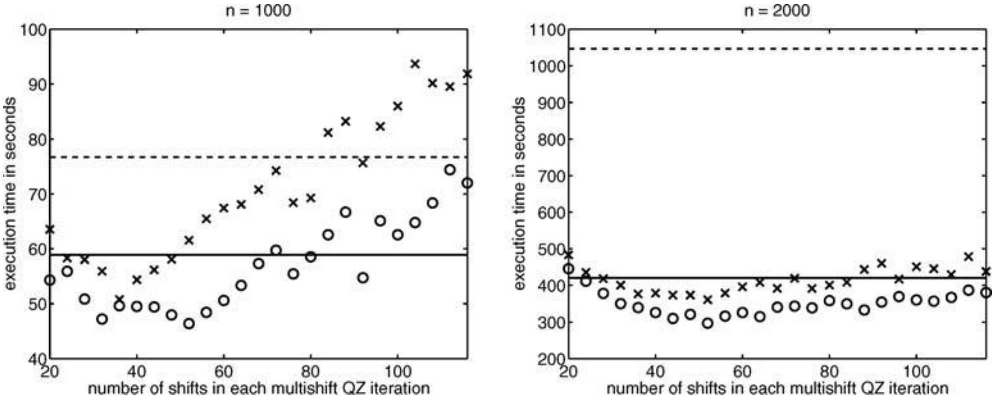


FIG. 7.1. *Random matrix pairs: Execution times on* `seth` *of* `DHGEQZ` *(dashed line),* `KDHGEQZ` *(solid line), and* `MULTIQZ` without *aggressive early deflation for* $n_s = 2$ *(crosses) and* $n_s = 4$ *(circles).*

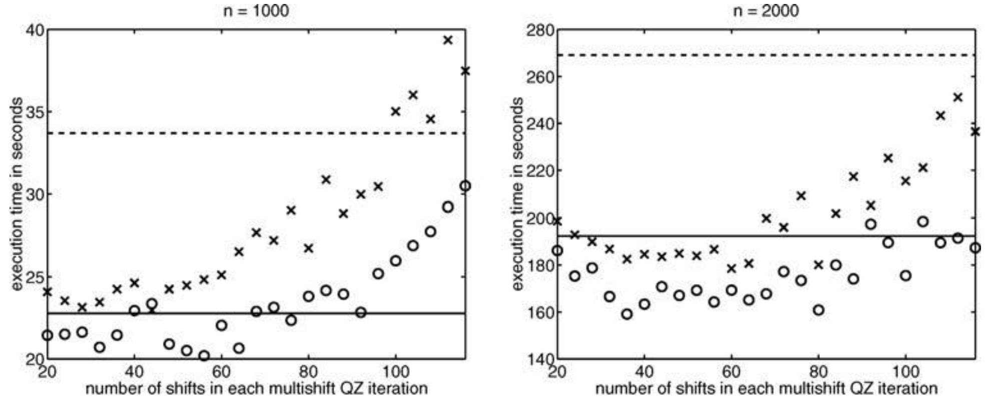


FIG. 7.2. *Random matrix pairs: Execution times on* `sarek` *of* `DHGEQZ` *(dashed line),* `KDHGEQZ` *(solid line), and* `MULTIQZ` without *aggressive early deflation for* $n_s = 2$ *(crosses) and* $n_s = 4$ *(circles).*

the measured execution times for the following implementations of the QZ algorithm:

| | |
|---|---|
| `DHGEQZ`: | LAPACK version 3.0 implementation as described in the original paper by Moler and Stewart [37] with some of the modifications proposed in [29, 47, 49]; see also section 2.1. |
| `KDHGEQZ`: | Blocked variant of `DHGEQZ`, developed by Dackland and Kågström [12]. |
| `MULTIQZ`: | Multishift QZ algorithm based on tightly coupled chains of tight bulges as described in section 3. |

The graphs in Figures 7.1 and 7.2 show the sensitivity of the measured execution times for $n = 1000$ and $n = 2000$ as a function of $m$, the degree of the multishift polynomial used in `MULTIQZ`, where $m$ is varying between 20 and 116 with step size 4. Note that in these and the following figures all results for a fixed value of $n$ are observations from a single random matrix pair. On `seth` it can be observed that the optimal time for `MULTIQZ` is significantly lower for both $n_s = 2$ and $n_s = 4$ than the time needed by `DHGEQZ` and `KDHGEQZ`. On `sarek` the gained savings are less substantial. In fact, for $n_s = 2$ and $n = 1000$ even with the optimal $m$, `MULTIQZ` requires slightly more execution time than `KDHGEQZ`.

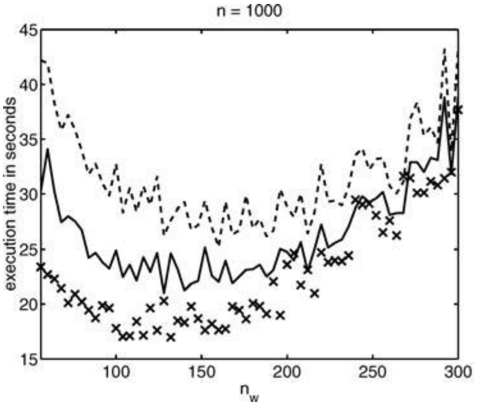


FIG. 7.3. *Random matrix pairs: Execution times on* `seth` *of* `DHGEQZ` with *aggressive early deflation (dashed line),* `KDHGEQZ` with *aggressive early deflation (solid line), and* `MULTIQZ` with *aggressive early deflation (crosses).*

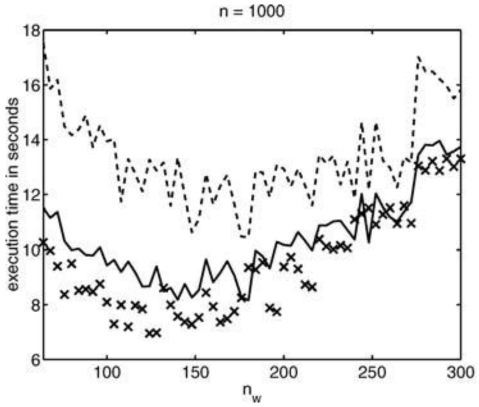


FIG. 7.4. *Random matrix pairs: Execution times on* `sarek` *of* `DHGEQZ` with *aggressive early deflation (dashed line),* `KDHGEQZ` with *aggressive early deflation (solid line), and* `MULTIQZ` with *aggressive early deflation (crosses).*

**7.2.2. Influence of $n_w$.** Similar results for the three implementations of the QZ algorithm with aggressive early deflation are displayed in Figures 7.3 and 7.4. The graphs show the sensitivity of the measured execution times with respect to $n_w$, the size of the deflation window, for $n = 1000$ and $n = 2000$. For `DHGEQZ` and `KDHGEQZ` aggressive early deflation has not been performed after each QZ iteration but only after every $m/2$ (double-shift) QZ iterations, meaning that an overall number of $m$ shifts is applied before each search for early deflations. For all three implementations, we have chosen the optimal value for $m$ in the set $\{20, 24, \dots, 116\}$. Moreover, we have set $n_s = 4$ for `MULTIQZ`. Again, it can be seen that `MULTIQZ` outperforms LAPACK's `DHGEQZ`, but is also faster than `KDHGEQZ`.

**7.2.3. Infinite eigenvalues.** To generate matrix pairs having large numbers of infinite eigenvalues, we generated Hessenberg-triangular matrix pairs $(H, T)$ in the same manner as in the previous two subsections and set each diagonal element of $T$ with probability 0.5 to zero. For $n = 2000$, this resulted in a matrix pair $(H, T)$ with roughly 1000 zero entries on the diagonal of $T$. Either implementation of the

Table 7.2

*Infinite eigenvalues: Execution times in seconds on* `seth` *and* `sarek` *of* `MULTIQZ` with *aggressive early deflation for a* 2000 × 2000 *random matrix pair with* 656 *infinite eigenvalues. The numbers shown in brackets correspond to the part (number between* 0 *and* 1*) of the execution time that was spent for deflating infinite eigenvalues.*

| Deflation strategy | `seth` | `sarek` |
|---|---|---|
| All $\infty$ eigenvalues at top left corner (unblocked) | 204.8 (0.70) | 71.6 (0.66) |
| All $\infty$ eigenvalues at top left corner (blocked) | 142.1 (0.56) | 59.8 (0.54) |
| All $\infty$ eigenvalues at nearest corner (unblocked) | 137.0 (0.60) | 43.5 (0.55) |
| All $\infty$ eigenvalues at nearest corner (blocked) | 91.6 (0.45) | 37.1 (0.44) |
| Necessary $\infty$ eigenvalues at nearest corner (unblocked) | 99.6 (0.11) | 43.9 (0.13) |
| Necessary $\infty$ eigenvalues at nearest corner (blocked) | 94.1 (0.05) | 42.8 (0.11) |

QZ algorithm detected 656 infinite eigenvalues. If only those infinite eigenvalues that correspond to nearly zero diagonal entries at the top left and bottom right corners of $T$ are deflated in the course of the QZ algorithm (see section 4.4), then a significant portion remains undetected. For example, when using this strategy together with `DHGEQZ` only 399 infinite eigenvalues were detected, which confirms the findings of Example 4.2. On the other hand, this strategy significantly lowers the time spent for dealing with infinite eigenvalues. This can be seen in the last two rows of Table 7.2, which lists execution times for various strategies used in `MULTIQZ` with $n_s = 4$ and the optimal values for $m$ and $n_w$ obtained from section 7.2.2. Note, however, that failing to detect infinite eigenvalues adversely affects the convergence of the QZ algorithm; the time spent for QZ iterations increases from 61 to 89 seconds on `seth`. We remark that the use of the windowing technique described in section 4.3 is denoted by "(blocked)" in Table 7.2. There are other interesting observations in the figures of this table. Deflating infinite eigenvalues at the nearest corner of the matrix $T$ (and not at only one corner as it is done in `DHGEQZ`) is a simple means to significantly lower the execution time. Roughly the same portion of time can be saved by using the windowing technique. The most efficient strategy, which detects all 656 infinite eigenvalues, is a combination of both techniques, deflation at the nearest corner combined with windowing.

**7.3. Aggressive early deflation at its best.** In exceptional cases, aggressive early deflation can have a dramatic positive effect on the computational time. Such a case are matrix pairs of the form (6.5), for which rarely any QZ iterations are needed to deflate eigenvalues. The graphs in Figure 7.5 show the measured execution times of the three implementations of the QZ algorithm with and without aggressive early deflation for $n = 600$ to 3000 (`seth`) or 4000 (`sarek`) with step size 200. For all examples we used $n_w = n - 1$.

We remark that since aggressive early deflation works so well, the time spent for (multishift) QZ iterations is negligible compared to the overall time. In fact, the timings for `DHGEQZ`, `KDHGEQZ`, and `MULTIQZ` with aggressive early deflation are virtually identical and orders of magnitude better than without early deflation. For example, for $n = 4000$ the time of `DHGEQZ` is reduced from nearly one hour to less than 7 seconds.

**7.4. Examples from applications.** The purpose of this section is to summarize the performance of the multishift QZ algorithm with aggressive early deflation for matrix pairs that arise from practically relevant applications. We have selected 16 matrix pairs from the Matrix Market collection [3], 6 matrix pairs from model reduction benchmark collections [11, 30], and 4 matrix pairs arising from the com-

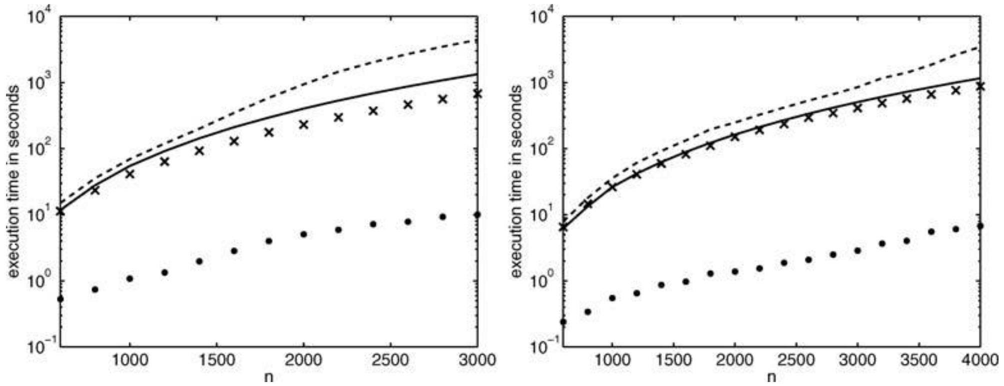


FIG. 7.5. *BBM/ADK example* [1, 6]*: Execution times (in logarithmic scale) on* `seth` *(left figure) and* `sarek` *(right figure) of* `DHGEQZ` *(dashed line),* `KDHGEQZ` *(solid line),* `MULTIQZ` *(crosses)* without *aggressive early deflation, and* `DHGEQZ/KDHGEQZ/MULTIQZ` with *aggressive early deflation (dots).*

putation of corner singularities of elliptic PDEs [38]. A more detailed description of the selected matrix pairs along with individual performance results can be found in Appendix A of the technical report version [22] of this paper. In the following, we summarize these results and sort the matrix pairs into groups according to their order $n$ as shown in the following table.

| Group | G1 | G2 | G3 | G4 |
|---|---|---|---|---|
| order | $n \in [485, 1000]$ | $n \in [1001, 1500]$ | $n \in [1501, 2000]$ | $n \in [2001, 3600]$ |
| #pairs | 6 | 8 | 5 | 7 |

Matrix pairs arising from applications differ in many aspects from random matrix pairs. An aspect which can particularly affect the performance of the QZ algorithm is that the matrix entries in most of the pairs from the Matrix Market collection differ wildly in magnitude. Bad scaling makes the performance of the QZ algorithm erratic and much less predictive than for random matrix pairs. For example, LAPACK's `DHGEQZ` requires 338 seconds for a $1900 \times 1900$ matrix pair arising from the discretized heat equation (see section A.19 in [22]) but less than 25 seconds for the $1922 \times 1922$ matrix pair consisting of the matrices BCSSTK26 and BCSSTM26 from the Matrix Market collection (see section A.12 in [22]). Balancing can remedy bad scaling but we have decided not to make use of it since this preprocessing step is by default turned off in most major software environments such as MATLAB.

We have tested `DHGEQZ`, `KDHGEQZ`, and `MULTIQZ` for all possible combinations of the parameters $n_w$ (aggressive early deflation window size), $m$ (number of shifts before each aggressive early deflation) and $n_s$ (number of shifts per bulge) satisfying $n_w \in \{40, 60, 80, \ldots, 400\}$, $m \in \{24, 32, 40, \ldots, 160\}$, and $n_s \in \{2, 4\}$. Due to the memory limitations of `seth`, all numerical experiments described in the following have only been performed on `sarek`. Also, to limit the variety of parameters, we have turned off the blocked algorithms for deflating infinite eigenvalues. Column 3 of Table 7.3 shows for each of the four groups the average times in seconds of `DHGEQZ` and `KDHGEQZ` without aggressive early deflation. The fourth column displays the average computing times for all three implementations with aggressive early deflation obtained by choosing $m$ and $n_w$ optimally for [illegible] matrix pair in the group. The fifth column displays similar times obtained by choosing $m$ and $n_w$ optimally to yield the best average performance for [illegible] matrix pairs together in each group. The corresponding choices

Table 7.3

*Application examples: Summary of measured execution times and choice of parameters $m$ and $n_w$ that give optimal average performance.*

| Group | Implementation | W/o agg. | Optimal | Ave. opt. | $m$ | $n_w$ |
|---|---|---|---|---|---|---|
| G1 | `DHGEQZ+AGG` | 8.86 | 4.61 | 5.03 | 24 | 120 |
| | `KDHGEQZ+AGG` | 7.07 | 3.62 | 3.78 | 24 | 100 |
| | `MULTIQZ(`$n_s=2$`)+AGG` | – | 3.36 | 3.45 | 24 | 60 |
| | `MULTIQZ(`$n_s=4$`)+AGG` | – | 3.40 | 3.56 | 24 | 40 |
| G2 | `DHGEQZ+AGG` | 55.5 | 23.7 | 26.5 | 56 | 180 |
| | `KDHGEQZ+AGG` | 41.4 | 17.2 | 18.9 | 40 | 140 |
| | `MULTIQZ(`$n_s=2$`)+AGG` | – | 15.3 | 16.5 | 72 | 160 |
| | `MULTIQZ(`$n_s=4$`)+AGG` | – | 15.1 | 16.3 | 88 | 180 |
| G3 | `DHGEQZ+AGG` | 130.3 | 53.6 | 56.4 | 48 | 220 |
| | `KDHGEQZ+AGG` | 89.9 | 36.5 | 38.8 | 72 | 200 |
| | `MULTIQZ(`$n_s=2$`)+AGG` | – | 30.3 | 30.7 | 56 | 200 |
| | `MULTIQZ(`$n_s=4$`)+AGG` | – | 27.5 | 30.0 | 88 | 200 |
| G4 | `DHGEQZ+AGG` | 479 | 157 | 170 | 48 | 340 |
| | `KDHGEQZ+AGG` | 271 | 97 | 104 | 40 | 220 |
| | `MULTIQZ(`$n_s=2$`)+AGG` | – | 80 | 85 | 72 | 220 |
| | `MULTIQZ(`$n_s=4$`)+AGG` | – | 115 | 124 | 80 | 220 |

of $m$ and $n_w$ are listed in columns 6 and 7, respectively. The difference between the figures of columns 4 and 5 is roughly 10%, which demonstrates that nearly optimal average performance can be obtained without having to optimize $m$ and $n_w$ for each matrix pair individually. Ideally, $m$ and $n_w$ should be chosen adaptively within the QZ algorithm, but it is not clear how such a strategy can be effectively realized.

On average, the multishift QZ algorithm with aggressive early deflation (`MULTIQZ +AGG`, $n_s = 2$) is between 2.6 and 6 times faster than the original LAPACK implementation (`DHGEQZ`). Surprisingly, the block version of Dackland and Kågström is, when equipped with aggressive early deflation (`KDHGEQZ+AGG`), only 10% to 20% slower than the multishift QZ algorithm. There is little justification for setting $n_s$, the number of shifts per bulge, to $n_s = 4$ in favor of $n_s = 2$, in contrast to the results for random matrix pairs.

We have also tested the backward stability of the new variants of the QZ algorithm by measuring the residual $\|(\hat{Q}^T A\hat{Z} - \hat{S}, \hat{Q}^T B\hat{Z} - \hat{T})\|_F$ of the computed Schur decomposition $(\hat{S}, \hat{T})$ as well as the orthogonality $\|\hat{Q}^T\hat{Q} - I\|_F, \|\hat{Z}^T\hat{Z} - I\|_F$ of the computed transformation matrices $\hat{Q}$ and $\hat{Z}$. The obtained results are of the same order as those obtained using the LAPACK implementation.

**8. Conclusions.** In this paper, we have developed new multishift variants of the QZ algorithm using advanced deflation techniques which significantly improve upon the performance compared to all existing implementations. It is planned that an implementation of our multishift QZ algorithm with aggressive early deflation is included in a coming release of LAPACK. The ideas presented here are currently applied to the development of a distributed memory QZ algorithm. Future work also includes the investigation of extending the described results to even more general versions of the QR algorithm, such as the periodic QR and QZ algorithms.

**9. Final remarks and acknowledgments.** The work presented in this article is based on preliminary results derived in [1, 31]. The authors are greatly indebted to Ralph Byers and David Watkins for several discussions on the odds and ends of multishift QR and QZ algorithms. The computational experiments in section 7 were

performed using facilities of the High Performance Computing Center North (HPC2N) in Umeå.

## REFERENCES

[1] B. Adlerborn, K. Dackland, and B. Kågström, *Parallel and blocked algorithms for reduction of a regular matrix pair to Hessenberg-triangular and generalized Schur forms*, in PARA 2002, J. Fagerholm et al., eds., Lecture Notes in Comput. Sci. 2367, Springer-Verlag, Berlin, 2002, pp. 319–328.

[2] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, 3rd ed., Software Environ. Tools 9, SIAM, Philadelphia, 1999.

[3] Z. Bai, D. Day, J. W. Demmel, and J. J. Dongarra, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems (Release* 1.0*)*, Technical report CS-97-355, Department of Computer Science, University of Tennessee, Knoxville, TN, 1997. Also available online from http://math.nist.gov/MatrixMarket.

[4] Z. Bai and J. W. Demmel, *On a block implementation of the Hessenberg multishift QR iterations*, Internat. J. High Speed Comput., 1 (1989), pp. 97–112.

[5] C. H. Bischof, B. Lang, and X. Sun, *A framework for symmetric band reduction*, ACM Trans. Math. Software, 26 (2000), pp. 581–601.

[6] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part* II: *Aggressive early deflation*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 948–973.

[7] K. Braman, R. Byers, and R. Mathias, *The multishift QR algorithm. Part* I: *Maintaining well-focused shifts and level* 3 *performance*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 929–947.

[8] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics Appl. Math. 14, SIAM, Philadelphia, 1995.

[9] R. Byers, C. He, and V. Mehrmann, *Where is the nearest non-regular pencil?*, Linear Algebra Appl., 285 (1998), pp. 81–105.

[10] R. Byers, *The descriptor controllability radius*, in Proceedings of the Conference on the Mathematical Theory of Networks and Systems, U. Helmke, R. Mennicken, and J. Saurer, eds., MTNS '93, Akademie Verlag, Berlin, 1994, pp. 85–88.

[11] Y. Chahlaoui and P. Van Dooren, *A Collection of Benchmark Examples for Model Reduction of Linear Time Invariant Dynamical Systems*, SLICOT working note 2002-2, WGS, 2002.

[12] K. Dackland and B. Kågström, *Blocked algorithms and software for reduction of a regular matrix pair to generalized Schur form*, ACM Trans. Math. Software, 25 (1999), pp. 425–454.

[13] J. W. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A - \lambda B$: Robust software with error bounds and applications.* I. *Theory and algorithms*, ACM Trans. Math. Software, 19 (1993), pp. 160–174.

[14] J. W. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil $A-\lambda B$: Robust software with error bounds and applications.* II. *Software and applications*, ACM Trans. Math. Software, 19 (1993), pp. 175–201.

[15] J. J. Dongarra, J. Du Croz, I. S. Duff, and S. Hammarling, *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[16] A. A. Dubrulle, *The Multishift QR Algorithm—Is it Worth the Trouble?*, Technical report TR 6320-3558, IBM Scientific Center, Palo Alto, CA, 1991.

[17] F.R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 1960.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[19] K. Goto and R. van de Geijn, *High-Performance Implementation of the Level-*3 *BLAS*, Technical report TR-2006-23, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 2006.

[20] E. Griepentrog and R. März, *Differential-Algebraic Equations and Their Numerical Treatment*, Teubner Texte zur Mathematik, Teubner-Verlag, Leipzig, 1986.

[21] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[22] B. Kågström and D. Kressner, *Multishift Variants of the QZ Algorithm with Aggressive Early Deflation*, Report UMINF-05.11, Department of Computing Science, Umeå University, Umeå, Sweden, 2005. Also appeared as LAPACK Working Note 173.

[23] B. Kågström, P. Ling, and C. F. Van Loan, *GEMM-based level* 3 *BLAS: Algorithms for the model implementations*, ACM Trans. Math. Software, 24 (1999), pp. 268–302.

[24] B. Kågström, P. Ling, and C. F. Van Loan, *GEMM-based level* 3 *BLAS: High-performance model implementations and performance evaluation benchmark*, ACM Trans. Math. Software, 24 (1999), pp. 303–316.

[25] B. Kågström and P. Poromaa, *Computing eigenspaces with specified eigenvalues of a regular matrix pair* $(A, B)$ *and condition estimation: Theory, algorithms and software*, Numer. Algorithms, 12 (1996), pp. 369–407.

[26] B. Kågström, *A direct method for reordering eigenvalues in the generalized real Schur form of a regular matrix pair* $(A, B)$, in Linear Algebra for Large Scale and Real-Time Applications (Leuven, 1992), M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., NATO Adv. Sci. Inst. Ser. E Appl. Sci. 232, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1993, pp. 195–218.

[27] B. Kågström, *Singular matrix pencils*, in Templates for the Solution of Algebraic Eigenvalue Problems, Z. Bai, J. W. Demmel, J. J. Dongarra, A. Ruhe, and H. van der Vorst, eds., Software Environ. Tools 11, SIAM, Philadelphia, 2000, pp. 260–277.

[28] L. Kaufman, *The LZ-algorithm to solve the generalized eigenvalue problem*, SIAM J. Numer. Anal., 11 (1974), pp. 997–1024.

[29] L. Kaufman, *Some thoughts on the QZ algorithm for solving the generalized eigenvalue problem*, ACM Trans. Math. Software, 3 (1977), pp. 65–75.

[30] J. G. Korvink and B. R. Evgenii, *Oberwolfach benchmark collection*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. C. Sorensen, eds., Lecture Notes Comput. Sci. Eng. 45, Springer-Verlag, Heidelberg, 2005, pp. 311–316.

[31] D. Kressner, *Numerical Methods and Software for General and Structured Eigenvalue Problems*, Ph.D. thesis, Institut für Mathematik, TU Berlin, Berlin, Germany, 2004.

[32] D. Kressner, *Block algorithms for reordering standard and generalized Schur forms*, LAPACK working note 171, September 2005. ACM Trans. Math. Software, to appear.

[33] D. Kressner, *On the use of larger bulges in the QR algorithm*, Electron. Trans. Numer. Anal., 20 (2005), pp. 50–63.

[34] V. N. Kublanovskaya, *AB-algorithm and its modifications for the spectral problems of linear pencils of matrices*, Numer. Math., 43 (1984), pp. 329–342.

[35] P. Kunkel and V. Mehrmann, *Differential-Algebraic Equations. Analysis and Numerical Solution*, EMS Publishing House, Zürich, Switzerland, 2006.

[36] B. Lang, *Effiziente Orthogonaltransformationen bei der Eigen- und Singulärwertzerlegung*, Habilitationsschrift, 1997.

[37] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[38] C. Pester, *CoCoS—Computation of Corner Singularities*, preprint SFB393/05-03, Technische Universität Chemnitz, Chemnitz, Germany, 2005.

[39] P. J. Rabier and W. C. Rheinboldt, *Nonholonomic Motion of Rigid Mechanical Systems from a DAE Viewpoint*, SIAM, Philadelphia, 2000.

[40] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.

[41] G. W. Stewart, *On the eigensystems of graded matrices*, Numer. Math., 90 (2001), pp. 349–370.

[42] T. Stykel, *Balanced truncation model reduction for semidiscretized Stokes equation*, Technical report 04-2003, Institut für Mathematik, TU Berlin, Berlin, Germany, 2003.

[43] C. Tischendorf, *Solution of Index-2-DAEs and Its Application in Circuit Simulation*, Dissertation, Humboldt-Univ. zu Berlin, Berlin, Germany, 1996.

[44] P. Van Dooren, *Algorithm* 590*: DSUBSP and EXCHQZ: Fortran subroutines for computing deflating subspaces with specified spectrum*, ACM Trans. Math. Software, 8 (1982), pp. 376–382.

[45] C. F. Van Loan, *Generalized Singular Values with Algorithms and Applications*, Ph.D. thesis, The University of Michigan, Ann Arbor, MI, 1973.

[46] R. C. Ward, *A Numerical Solution to the Generalized Eigenvalue Problem.*, Ph.D. thesis, University of Virginia, Charlottesville, VA., 1974.

[47] R. C. Ward, *The combination shift QZ algorithm*, SIAM J. Numer. Anal., 12 (1975), pp. 835–853.

[48] R. C. Ward, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 141–152.

[49] D. S. Watkins and L. Elsner, *Theory of decomposition and bulge-chasing algorithms for the generalized eigenvalue problem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 943–967.

[50] D. S. Watkins, *Shifting strategies for the parallel QR algorithm*, SIAM J. Sci. Comput., 15 (1994), pp. 953–958.

[51] D. S. Watkins, *Forward stability and transmission of shifts in the QR algorithm*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 469–487.

[52] D. S. Watkins, *The transmission of shifts and shift blurring in the QR algorithm*, Linear Algebra Appl., 241/243 (1996), pp. 877–896.

[53] D. S. Watkins, *Performance of the QZ algorithm in the presence of infinite eigenvalues*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 364–375.

[54] J. H. Wilkinson, *Kronecker's canonical form and the QZ algorithm*, Linear Algebra Appl., 28 (1979), pp. 285–303.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 228–244

# DIAGONAL MARKOWITZ SCHEME WITH LOCAL SYMMETRIZATION*

PATRICK R. AMESTOY†, XIAOYE S. LI‡, AND ESMOND G. NG‡

**Abstract.** We describe a fill-reducing ordering algorithm for sparse, nonsymmetric LU factorizations, where the pivots are restricted to the diagonal and are selected greedily. The ordering algorithm uses only the structural information. Most of the existing methods are based on some type of symmetrization of the original matrix. Our algorithm exploits the nonsymmetric structure of the given matrix as much as possible. The new algorithm is thus more complex than classical symmetric orderings, but we show that our algorithm can be implemented in space bounded by the number of nonzero entries in the original matrix, and has the same time complexity as the analogous algorithms for symmetric matrices. We provide numerical experiments to demonstrate the ordering quality and the runtime of the new ordering algorithm.

**Key words.** sparse nonsymmetric matrices, linear equations, ordering methods

**AMS subject classifications.** 65F05, 65F50

**DOI.** 10.1137/050637315

## 1. Introduction.

We consider the direct solution of sparse linear equations $\mathbf{Ax} = \mathbf{b}$ using Gaussian elimination, where $\mathbf{A}$ is an $n \times n$ nonsymmetric sparse matrix. A major difficulty with nonsymmetric matrices is that they are rarely diagonally dominant, which means that during numerical factorization one must compromise fill-in reduction with numerical stability. Many nonsymmetric solvers deal with this situation using the [illegible], which includes an [illegible] phase, a [illegible] phase, and a [illegible] phase [2, 6, 16]. Iterative refinements may be included in the triangular solution. The analysis phase includes a (numerical) preprocessing of the matrix and a symbolic phase that builds the computational graph for the numerical factorization phase. An advantage of the three-phase approach lies in its ability to anticipate the choice of the next pivot, which decouples the analysis from factorization and makes parallelization of numerical factorization easier. Therefore, it is a very important class of methods on high performance computers. In this context, it has been observed in [3] that it is critical to put numerically large entries on the diagonal during the preprocessing phase to limit the scope of numerical pivoting during numerical factorization. One may then want to preserve this diagonal during sparsity reordering. That is, only a symmetric permutation is allowed afterward. One common practice to obtain such an ordering is to apply a symmetric ordering algorithm, either minimum-degree or nested-dissection variant, to the symmetrized pattern of $A + A^T$. Such reordering algorithms do not exploit the fact that during factorization the solvers can exploit the asymmetry of the permuted matrix.

---

*Received by the editors August 1, 2005; accepted for publication (in revised form) by M. Benzi July 31, 2006; published electronically January 12, 2007. This work was supported in part by the Director, Office of Advanced Scientific Computing Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098, and in part by NSF-INRIA grant NSF-INT-0003274.

http://www.siam.org/journals/simax/29-1/63731.html

†ENSEEIHT-IRIT, 2 rue Camichel, 31071 Toulouse, France (amestoy@enseeiht.fr). Part of the work of this author was performed while he was on a sabbatical visit to Lawrence Berkeley National Laboratory.

‡Lawrence Berkeley National Laboratory, One Cyclotron Road, Mail Stop 50F-1650, Berkeley, CA 94720 (xsli@lbl.gov, egng@lbl.gov).

In this article, we propose a new symmetric ordering algorithm, working directly on $A$ and exploiting the nonsymmetric structure of $A$, to compute a "good" symmetric permutation of $A$. It is based on greedy heuristics that preserve the large diagonal entries and at the same time take into account the asymmetry of the matrix. In the symmetric case, the minimum-degree algorithm is a very effective greedy heuristic for fill-in reduction. By using the quotient graph elimination model [12, 13] and the approximate degree updates [1], the minimum-degree algorithm can be implemented very efficiently both in time and space. The nonsymmetric variant of minimum-degree was actually discovered earlier and was named after Markowitz [19], in which the "degree" of a vertex is the product of the row count and column count (known as the Markowitz count). But the original Markowitz algorithm is asymptotically slower than the minimum-degree algorithm, mainly due to the lack of a concise quotient graph model. A theoretical advancement was made by Pagallo and Maulino [22], who extended the quotient graph idea for symmetric matrices to the nonsymmetric case by introducing the [illegible] and showed that the bipartite quotient graph model can be implemented in space bounded by the size of **A**. But timewise, using only the quotient graph model does not lead to an ordering algorithm that is as fast as the minimum-degree algorithm. This is because the lengths of the reachable paths to be searched when updating the Markowitz counts are not bounded. One main contribution of our work is the introduction of a [illegible] mechanism that bounds the lengths of the reachable paths as in the symmetric case while capturing most of the asymmetry in the matrix. A secondary contribution is to adapt and extend the metrics to select pivots based on approximate degree [1] to metrics based on approximate Markowitz count and deficiency [24, 20]. Indeed, in our context all metrics have to anticipate the effect that local symmetrization would have on the pivot to be selected. Our algorithm has the same asymptotic complexity as the minimum-degree algorithm, both in space and in time.

The remainder of the paper is organized as follows. In section 2, we first briefly introduce the bipartite quotient graph notation and properties. We then present the local symmetrization technique and describe our new ordering algorithm. In particular, we discuss how to update the quotient graph and how to compute metrics to select pivots within this framework. Section 3 describes the numerical experiments we have performed and analyzes the effect of the new ordering algorithm on the multifrontal code `MA41_UNS` [2, 6]. Section 4 provides a summary of this research.

## 2. Diagonal Markowitz with local symmetrization.

This section presents the algorithmic ingredients of our new Markowitz ordering framework. We show that the Markowitz algorithm can be implemented as efficiently as the approximate minimum-degree algorithm by using bipartite quotient graphs, the local symmetrization scheme, and the metrics based on approximate row and column degrees.

### 2.1. The Markowitz criterion.

The Markowitz ordering algorithm [19] has been used successfully in general-purpose solvers [11]. This local greedy strategy can be described succinctly as follows. After $k$ steps of Gaussian elimination, let $r_i^k$ (resp., $c_j^k$) denote the number of nonzero entries in row $i$ (resp., column $j$) of the remaining $(n-k) \times (n-k)$ submatrix. The (structural) Markowitz criterion is to select, as the next pivot, a nonzero entry $a_{ij}^k$ from the remaining submatrix that has the minimum Markowitz count $(r_i^k - 1) \times (c_j^k - 1)$. This attempts to minimize an upper bound on the amount of fill-in generated at step $k+1$. Note that, in our context, we want to restrict the pivot selection to the diagonal of the remaining submatrix. This restriction of the

Markowitz scheme will be referred to as the [illegible] scheme.

The simple rule above for choosing the next pivot does not immediately render an efficient implementation, because it requires updating the sparsity pattern of the remaining submatrix at each step, which may generate fill-in. From the development of the minimum-degree algorithm, which can be considered as a symmetric variant of Markowitz algorithm, we learned that by using the quotient graph elimination model [13], the algorithm can be implemented in space bounded by the size of the original matrix rather than that of the filled matrix. This is the so-called [illegible] property and is very much desirable in an efficient ordering algorithm. Pagallo and Maulino [22] extended the quotient graph model by using [illegible] to model the nonsymmetric elimination and showed that this model indeed has the in-place property. Now we briefly review this concept and illustrate how we can use and modify this model to design our ordering algorithm.

**2.2. Bipartite quotient graphs.** Let $\mathbf{A}$ be a nonsymmetric $n \times n$ matrix. The nonzero pattern of $\mathbf{A}$ can be represented by a bipartite graph $G = (V_r, V_c, E)$, where $V_r$ and $V_c$ are the sets of row and column vertices, respectively. For a row vertex $r_i \in V_r$ and a column vertex $c_j \in V_c$, an edge $(r_i, c_j) \in E$ exists if and only if $a_{ij} \neq 0$. Let $G^0 = (V_r^0, V_c^0, E^0)$ be the same as $G$. We use a bipartite graph $G^k = (V_r^k, V_c^k, E^k)$ to represent the nonzero pattern of the remaining submatrix after $k$ steps of Gaussian elimination. Assuming pivots are chosen from the main diagonal, at step $k$, the transformation from $G^{k-1}$ to $G^k$ is based on the following elimination rule. Suppose the $k$th pivot node $(r_p, c_p)$, $p \geq k$, is selected for elimination. The vertex sets become $V_r^k = V_r^{k-1} \backslash \{r_p\}$ and $V_c^k = V_c^{k-1} \backslash \{c_p\}$. The edge set $E^k$ is derived from $E^{k-1}$ by deleting the edges incident on $c_p$ and $r_p$ and adding edges $(r_i, c_j)$ for all $r_i$ and $c_j$ that are adjacent to $c_p$ and $r_p$, respectively. This creates a fully connected bipartite subgraph (a [illegible] in the symmetric analogue). We may refer to this as a [illegible], or [illegible] in short.

We now briefly review the symmetric quotient graph elimination model. The main idea is to use a compact representation to implicitly store the subgraph induced by the vertices that have been eliminated. Suppose $G_s$ is a undirected graph corresponding to a sparse symmetric matrix. Let $S$ denote the subset of vertices in $G_s$ that have been eliminated. Consider the subgraph $G_s(S)$ induced by $S$ in $G_s$. In the quotient graph model, each [illegible][1] in $G(S)$ will be represented by a single "supervertex." As a result, any path in $G_s$ from a vertex $i \notin S$ to a vertex $j \notin S$ through $S$ corresponds to a path through at most one supervertex in $S$. The set of vertices adjacent to $i$ in the remaining filled subgraph is given precisely by the reachable set of $i$ through $S$. See [13] for details.

We now describe the nonsymmetric elimination process using the bipartite quotient graph model. We will use calligraphic letters to denote the sets associated with the bipartite quotient graph. Let $\mathcal{G}^k$ denote the bipartite quotient graph which represents the structure of the reduced submatrix after $k$ steps of Gaussian elimination, and define $\mathcal{G}^0 = G^0$. When there is no ambiguity, we will omit superscript $k$. Both row and column vertices are partitioned into two sets: the set of uneliminated vertices referred to as [illegible] and the set of eliminated vertices referred to as [illegible]. That is, $\mathcal{G} = (\mathcal{V}_r \cup \bar{\mathcal{V}}_r, \mathcal{V}_c \cup \bar{\mathcal{V}}_c, \mathcal{E} \cup \bar{\mathcal{E}})$. Members of $\mathcal{V}_r$ ($\mathcal{V}_c$) will be referred to as [illegible] (to distinguish them from the row vertices in $V_r$ ($V_c$)), while members of $\bar{\mathcal{V}}_r$ ($\bar{\mathcal{V}}_c$) will be referred to as [illegible]. The edge set $\mathcal{E}$ contains the

---

[1] A connected component is a graph in which there is a path between every pair of vertices.

edges between row (column) and column (row) variables. The edge set $\bar{\mathcal{E}}$ contains the edges between row (column) variables and column (row) elements, as well as the edges between row elements and column elements. An eliminated pivot $e = (r_e, c_e)$ has two vertices $r_e \in \bar{\mathcal{V}}_r$ and $c_e \in \bar{\mathcal{V}}_c$ referred to as a [illegible]. Similarly an uneliminated pivot entry (a diagonal entry in the reduced matrix) $i = (r_i, c_i)$ will be referred to as a [illegible]. A nonzero entry $(r_i, c_j)$ exists in the factors if and only if there exists a path of the form $r_i \to c_{e_1} \to r_{e_1} \ldots \to c_{e_l} \to r_{e_l} \to c_j$, where $e_i = (r_{e_i}, c_{e_i}), 1 \le i \le l$, are the coupled elements associated with the pivots already eliminated [23]. Therefore, following such paths, we can determine the nonzero entries of any row $i$ or column $j$ in the reduced submatrix.

Let $\mathcal{A}_{i*}$ be the set of column variables adjacent to row variable $r_i$ in $\mathcal{G}$ which have never been modified after $k$ steps of elimination. $\mathcal{A}_{*j}$ is defined similarly for column variable $c_j$. For each row variable $r_i$ and column variable $c_j$, define the element adjacency lists:

$$\mathcal{R}_i \equiv \{e = (r_e, c_e) \,:\, (r_i, c_e) \in \bar{\mathcal{E}}\} \subseteq \bar{\mathcal{V}}_c, \text{ the set of coupled elements adjacent to } r_i\,,$$
$$\mathcal{C}_j \equiv \{e = (r_e, c_e) \,:\, (r_e, c_j) \in \bar{\mathcal{E}}\} \subseteq \bar{\mathcal{V}}_r, \text{ the set of coupled elements adjacent to } c_j\,.$$

The adjacency lists of variables in the current bipartite quotient graph are then defined as

$$\mathcal{U}_i \equiv Adj^{row}_{\mathcal{G}}(r_i) = \mathcal{A}_{i*} \cup \mathcal{R}_i\,, \tag{1}$$
$$\mathcal{L}_j \equiv Adj^{col}_{\mathcal{G}}(c_j) = \mathcal{A}_{*j} \cup \mathcal{C}_j\,. \tag{2}$$

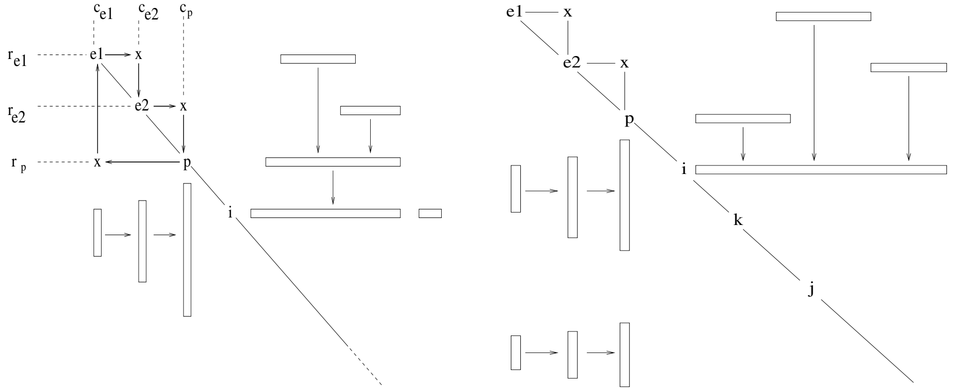
For each coupled element $e = (r_e, c_e)$ define the variable adjacency lists:

$$\mathcal{L}_e \equiv \{r_i \,:\, (r_i, c_e) \in \bar{\mathcal{E}}\} \subseteq \mathcal{V}_r, \text{ the set of row variables adjacent to } c_e\,,$$
$$\mathcal{U}_e \equiv \{c_j \,:\, (r_e, c_j) \in \bar{\mathcal{E}}\} \subseteq \mathcal{V}_c, \text{ the set of column variables adjacent to } r_e\,.$$

In other words, $\mathcal{L}_e$ and $\mathcal{U}_e$ are, respectively, the sets of row and column vertices in the biclique induced after elimination of the coupled element $e$.

Now, suppose a pivot $p = (r_p, c_p)$, $p \ge k$, is chosen to be eliminated next. If there exists a cycle of the form $r_p \to c_{e_1} \to r_{e_1} \ldots \to c_{e_l} \to r_{e_l} \to c_p \to r_p$ $(e_i \le k, 1 \le i \le l)$ (referred to as a [illegible] in [22]), then $\mathcal{L}_{e_i} \subseteq \mathcal{L}_p$ and $\mathcal{U}_{e_i} \subseteq \mathcal{U}_p$ for all $i$. Hence, except for $(r_p, c_p)$, the other coupled elements in the cycle are no longer needed. See Figure 1(a) for an illustration. When updating the quotient graph, we can coalesce the coupled elements in the cycle into a single "supervertex," using the last element $p$ as the representative vertex and removing the other elements and the incident edges. This process will be referred to as [illegible].

The transformation from bipartite quotient graph $\mathcal{G}^{k-1}$ to $\mathcal{G}^k$ at step $k$ is carried out as follows. We search in the subgraph of $\mathcal{G}^{k-1}$ induced by $\bar{\mathcal{V}}_r^{k-1} \cup \bar{\mathcal{V}}_c^{k-1}$ for cycles that include the pivot $(r_p, c_p)$. We then perform the element absorptions and form the new adjacency lists $\mathcal{L}_p$ and $\mathcal{U}_p$. The structure of a column $k$ in the reduced submatrix, $\mathbf{L}_{*k}$, can be determined very easily using $\mathcal{G}^{k-1}$: $\ell_{ik} \ne 0$ if and only if $r_i$ is reachable from $c_k$ through the coupled elements in $\mathcal{G}^{k-1}$. The structure of $\mathbf{U}_{k*}$ can be determined in a similar way. The biclique introduced by the current pivot is then used to prune the edges in $\mathcal{E}^k$. This process will be referred to as [illegible]. From the variable pruning process it results that $(r_i, c_j) \in \mathcal{E}^k$ if and only if $(r_i, c_j) \in E$ and entry $a_{i,j}$ of the original matrix has not been modified during steps 1 through $k$ of the elimination. It was proved that using this scheme, the in-place property is maintained



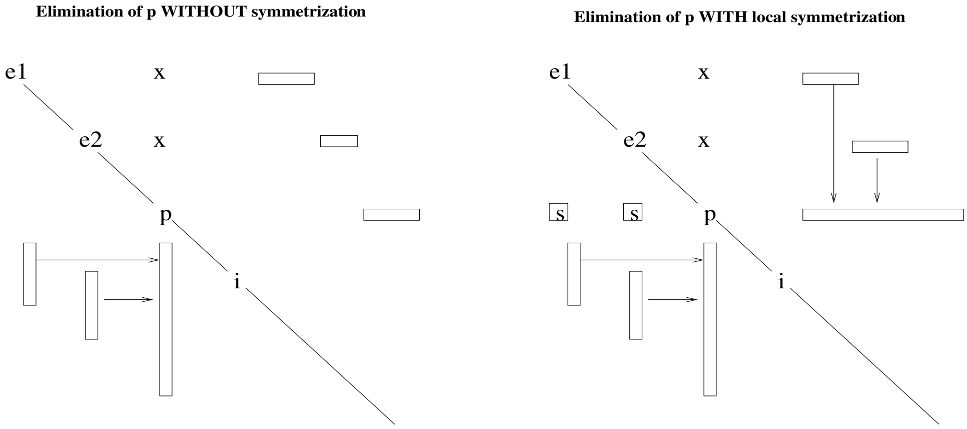
(a) Quotient graph with a cycle. (b) Quotient graph without cycle.

FIG. 1. (a) *Quotient graph with a cycle:* $r_p \rightarrow c_{e_1} \rightarrow r_{e_1} \rightarrow c_{e_2} \rightarrow r_{e_2} \rightarrow c_p$. (b) *Quotient graph without cycle.*

for each $\mathcal{G}^k$ [22]. But unlike the symmetric case, here, computing the reachable sets can be very expensive, because the length of the search path is not bounded by two. (In fact, it can be as long as $|\bar{\mathcal{V}}_r \cup \bar{\mathcal{V}}_c| + 1$ if no cycle is found.) This is illustrated by the example in Figure 1(b). There is only a simple path between $p$ and elements $e_1$ and $e_2$, which results in $\mathcal{L}_{e_1} \subseteq \mathcal{L}_{e_2} \subseteq \mathcal{L}_p$. However, $\mathcal{U}_{e_1} \nsubseteq \mathcal{U}_p$ and $\mathcal{U}_{e_2} \nsubseteq \mathcal{U}_p$. For an uneliminated variable $i$ such that $r_i \in \mathcal{L}_p \cap \mathcal{L}_{e_1} \cap \mathcal{L}_{e_2}$, all the column variables in $\mathcal{U}_p \cup \mathcal{U}_{e_2} \cup \mathcal{U}_{e_1}$ need to be included in $\mathbf{U}_{i*}$. In an in-place algorithm, one must store $r_i$ only in $\mathcal{L}_{e_1}$, and then via the path $c_{e_1} \rightarrow r_{e_1} \rightarrow c_{e_2} \rightarrow r_{e_2} \rightarrow c_p$ one can deduct that $\mathbf{U}_{i*}$ should contain the union $\mathcal{U}_p \cup \mathcal{U}_{e_2} \cup \mathcal{U}_{e_1}$. We also note that, although $\mathcal{L}_{e_1} \subseteq \mathcal{L}_{e_2} \subseteq \mathcal{L}_p$, $\mathcal{L}_{e_1}$ alone may be required to build $\mathbf{L}_{*j}$ for any $j$ such that $c_j \in \mathcal{U}_{e_1}$ and $c_j \notin \mathcal{U}_{e_2} \cup \mathcal{U}_p$. This means that $\mathcal{L}_{e_1}$ should not be absorbed into $\mathcal{L}_p$. Furthermore, if one considers a variable $k$ such that $c_k \in \mathcal{U}_p$ but $c_k \notin \mathcal{U}_{e_1} \cup \mathcal{U}_{e_2}$, then $\mathcal{L}_p$ will need to be included in $\mathcal{L}_k$. If we maintain the in-place property, the entries belonging to both $\mathcal{L}_{e_i}$, $i = 1, 2$, and $\mathcal{L}_p$ are stored only in $\mathcal{L}_{e_i}$, then we must be able to reach $e_i$, $i = 1, 2$, through a path starting at $p$: $c_p \rightarrow r_{e_2} \rightarrow c_{e_2} \rightarrow r_{e_1}$.

**2.3. Local symmetrization.** To avoid the long search path in a truly nonsymmetric algorithm, we have designed a relaxed diagonal Markowitz scheme. Figure 2 illustrates such a relaxation. The entry marked $\boxed{\text{s}}$ shows an artificial nonzero introduced to symmetrize only a [illegible] part of the matrix. In the example, we assume that $(r_p, c_p)$ is the current pivot, and $\mathcal{R}_p = \emptyset$ and $\mathcal{C}_p = \{e_1, e_2\}$. We also assume that $\mathcal{U}_{e_1} \nsubseteq \mathcal{U}_p$ and $\mathcal{U}_{e_2} \nsubseteq \mathcal{U}_p$. For the sake of clearness, we have assumed that $\mathcal{U}_{e_1} \cap \mathcal{U}_p = \emptyset$ and $\mathcal{U}_{e_2} \cap \mathcal{U}_p = \emptyset$. In order to obtain the row structure $\mathbf{U}_{i*}$, where $r_i \in \mathcal{L}_{e_1} \cap \mathcal{L}_{e_1} \cap \mathcal{L}_p$, $\mathcal{R}_i$ must contain elements $e_1$, $e_2$, and $p$. In other words, all the variables in $\mathcal{U}_{e_1} \cup \mathcal{U}_{e_2} \cup \mathcal{U}_p$ should be included in $\mathbf{U}_{i*}$. With symmetrization (shown on the right part of Figure 2), we pretend that $\mathcal{R}_p = \{e_1, e_2\}$ and $\mathcal{C}_p = \{e_1, e_2\}$. Therefore, $\mathcal{U}_{e_1} \subseteq \mathcal{U}_p$ and $\mathcal{U}_{e_2} \subseteq \mathcal{U}_p$. Hence, the coupled element $p$ can absorb the coupled elements $e_1$ and $e_2$. As a result, we now need only the adjacency lists of $r_p$ and $c_p$ to get the adjacency lists of $r_i$ and $c_i$. This eliminates the need to keep the adjacency lists of $r_{e_1}$, $r_{e_2}$, $c_{e_1}$, and $c_{e_2}$.

In summary, the local symmetrization works as follows. Suppose the current pivot



**Elimination of p WITHOUT symmetrization**

**Elimination of p WITH local symmetrization**

FIG. 2. *Illustration of local symmetrization.*

is $(r_p, c_p)$. The adjacency lists $\mathcal{U}_p$ and $\mathcal{L}_p$ are computed by

$$\mathcal{U}_p = \left( \mathcal{A}_{p*} \cup \bigcup_{e \in \mathcal{R}_p} \mathcal{U}_e \cup \bigcup_{e \in \mathcal{C}_p} \mathcal{U}_e \right) \setminus \{c_p\}, \tag{3}$$

$$\mathcal{L}_p = \left( \mathcal{A}_{*p} \cup \bigcup_{e \in \mathcal{C}_p} \mathcal{L}_e \cup \bigcup_{e \in \mathcal{R}_p} \mathcal{L}_e \right) \setminus \{r_p\}. \tag{4}$$

The third terms in the unions result from the local symmetrization. The adjacency lists in the bipartite quotient graph (see (1) and (2)) of all the row (column) variables in the adjacency lists of the newly formed coupled element $p$ should then be updated. All the row and column elements in $\mathcal{R}_p \cup \mathcal{C}_p$ are absorbed by the coupled element $p$. Therefore, if $(r_e, c_e)$ is such an absorbed element, then $r_e$ $(c_e)$ will be replaced by $r_p$ $(c_p)$ each time it appears in an edge of $\bar{\mathcal{E}}$ and will be excluded from the quotient graph together with $\mathcal{L}_e$ $(\mathcal{U}_e)$. Furthermore, because of local symmetrization, more variable pruning can be performed. Let $i = (r_i, c_i)$ be a coupled variable (diagonal entry in the reduced matrix) such that $r_i \in \mathcal{L}_p$ and $c_i \notin \mathcal{U}_p$. We can anticipate local symmetrization between $i$ and the coupled element $p$ to prune all the row variables in $\mathcal{A}_{*i}$ that belong to $\mathcal{L}_p$. Entries in $\mathcal{A}_{i*}$ can also be pruned in a similar way (even if $r_i \notin \mathcal{L}_p$).

Our relaxation mechanism will be referred to as [illegible], because the symmetrization is applied to only the local part of the graph involving only those row and column elements adjacent to $c_p$ and $r_p$. Globally, the nonzero structure generally still remains nonsymmetric (the index sets $\{k : r_k \in \mathbf{L}_{*i}\}$ and $\{k : c_k \in \mathbf{U}_{i*}\}$ are different). By construction, the length of a search path is bounded by three. In essence, we trade off some amount of asymmetry and space (because some zero entries may be stored) with a much faster search algorithm. We show in Theorem 2.1 that although the local symmetrization may introduce extra (zero) entries in the factors with respect to a pure nonsymmetric scheme (see Figure 2), it leads to an in-place algorithm.

THEOREM 2.1. [illegible] $v$ [illegible] $\mathcal{G}^k$ [illegible] $\mathcal{A}^k_{v*}$ [illegible] $\mathcal{A}^k_{*v}$ [illegible] $v$ [illegible]

[illegible] $k$ [illegible] $2 \leq k \leq n$ $|\mathcal{A}^k_{v*}| + |\mathcal{R}^k_v| \leq |\mathcal{A}^{k-1}_{v*}| + |\mathcal{R}^{k-1}_v| \leq |\mathcal{A}^0_{v*}|$ [illegible] $|\mathcal{A}^k_{*v}| + |\mathcal{C}^k_v| \leq |\mathcal{A}^{k-1}_{*v}| + |\mathcal{C}^{k-1}_v| \leq |\mathcal{A}^0_{*v}|$

[illegible]. We focus on the row structures in this proof. The proof for the column structures is similar. We prove this theorem by induction. By construction, $\mathcal{R}^0_i = \emptyset$, so $|\mathcal{A}^0_{i*}| + |\mathcal{R}^0_i| = |\mathcal{A}^0_{i*}|$. Suppose at the $k$th step of elimination that $(r_p, c_p)$ is selected as the pivot. We first build $\mathcal{U}^k_p$ using (3). The entries in $\mathcal{U}^k_p$ either come from the original matrix $\mathcal{A}_{p*}$ or from the entries in $\mathcal{U}_e$ such that $e \in \mathcal{R}_p \cup \mathcal{C}_p$. Because of local symmetrization, the coupled element $e$ will be absorbed by $p$, and the space of $\mathcal{U}_e$ can be used by $\mathcal{U}^k_p$ to store the new entries from $\mathcal{U}_e$. To take into account the fill-in we have to update the adjacency lists of all variables adjacent to the pivot. We focus on the row structures and thus consider the updating of $\mathcal{U}^k_i$ for $r_i \in \mathcal{L}^k_p$ using (1). By construction, all the entries in $\mathcal{U}^k_p \cap \mathcal{A}^{k-1}_{i*}$ are pruned from $\mathcal{A}^{k-1}_{i*}$, showing that $|\mathcal{A}^k_{i*}| \leq |\mathcal{A}^{k-1}_{i*}|$, and $c_p$ is added to $\mathcal{R}^k_i$. Now we consider the size of $\mathcal{R}^k_i$. If $r_i \in \mathcal{L}^k_p$, then there exists a coupled element $e_j = (r_j, c_j)$ in the supervertex $p = (r_p, c_p)$ such that $(r_i, c_j)$ is in the original graph. Since $(r_i, c_j)$ has been pruned and $c_j$ cannot belong to any other supervertex, then we have $|\mathcal{A}^k_{i*}| + |\mathcal{R}^k_i| \leq |\mathcal{A}^{k-1}_{i*}| + |\mathcal{R}^{k-1}_i|$. This concludes our proof of the in-place property of the algorithm. ☐

COROLLARY 2.2. [illegible] $\mathcal{G}^k$ [illegible] $\mathcal{G}^0$ [illegible] $|\mathcal{E}^k \cup \bar{\mathcal{E}}^k| \leq |\mathcal{E}^{k-1} \cup \bar{\mathcal{E}}^{k-1}| \leq |\mathcal{E}^0|$, [illegible] $2 \leq k \leq n$

Theorem 2.1 implies that the in-place property holds for the row adjacency lists and for the column adjacency lists so that we could have an in-place implementation while keeping two separate lists to store entries in rows and in columns at each step of the elimination.

We will call our relaxed scheme [illegible] (`DMLS`). We now illustrate its main properties with an example. In Figure 3, we apply the `DMLS` algorithm assuming that pivots are in the natural ordering. The matrix on the right is the structure of the LU factors. The elimination tree [18] built by the `DMLS` algorithm is shown in Figure 4. Each node of the tree corresponds to the elimination of a pivot. The nonsymmetric frontal matrix of each node corresponds to the structure of $\mathcal{U}_p$ and $\mathcal{L}_p$ as defined by (3) and (4). The dark area corresponds to the entries in the reduced matrix updated during the node elimination (i.e., the nonsymmetric contribution block sent by one node to its parent).

**Original Matrix**

| | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| **1** | X | X | X | |
| **2** | | X | | |
| **3** | | | X | X |
| **4** | X | | | X |

**LU factors with DMLS**

| | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| **1** | X | X | X | |
| **2** | | X | **S** | |
| **3** | | | X | X |
| **4** | X | **F** | **F** | X |

FIG. 3. *Illustration of the fill-ins introduced by the* `DMLS` *algorithm.* **F** *corresponds to the normal fill-in when eliminating pivot* $(1,1)$*.* **S** *corresponds to the fill-in due to the local symmetrization when eliminating pivot* $(2,2)$*.*

At the first step, pivot $(1,1)$ is eliminated resulting in two fill-ins (**F** in positions $(4,2)$ and $(4,3)$). In the quotient graph $\mathcal{G}^1$, these fill-ins are implicitly represented by removing $r_1$ from $\mathcal{A}_{*2}$ and $\mathcal{A}_{*3}$ and adding $r_1$ to $\mathcal{C}_2$ and $\mathcal{C}_3$. Note that at this step there is no symmetrization of the column and row adjacency lists of $r_1$ and $c_1$, which otherwise would result in a completely full reduced matrix. When eliminating

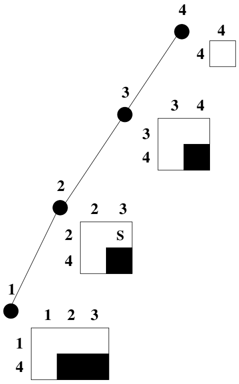


FIG. 4. *Elimination tree built by the* `DMLS` *algorithm applied to the matrix of Figure* 3.

pivot $(2,2)$, since $r_1 \in \mathcal{C}_2$ and $c_1 \notin \mathcal{R}_2$, local symmetrization is applied, and when computing $\mathcal{U}_2$ by (3) entry $\mathbf{S}$ in position $(2,3)$ is added to the quotient graph $\mathcal{G}^2$ (i.e., the coupled element 1 is absorbed by 2, and $r_2$ is added to $\mathcal{C}_3$). One should note that entry $(2,1)$ is only [illegible] considered as nonzero and is never added in the LU factors. Similarly, when eliminating pivot $(3,3)$ at the next step, only the effect of adding entry $(3,2)$, by symmetrization of $(2,3)$, on the structure of the column $\mathcal{L}_3$ is considered (it happens to have no effect in our example). Even if entry $(3,2)$ is not effectively stored and has no effect on the size of the factors, it still has an effect on the structure of the dependency graph, as shown in Figure 4. The fact that pivot 3 can absorb the coupled element 2 because of the artificial $(3,2)$ nonzero entry also means that node 3 in Figure 4 becomes the unique parent of node 2 in the dependency graph, which in turn becomes a tree (or forest when the matrix is reducible). It is also interesting to note that, since entry $(2,3)$ ($\mathbf{S}$ in the figure) is considered nonzero, column 3 is added to the frontal matrix of node 2. But entry $(4,3)$ will not be modified during elimination of pivot $(2,2)$, because entry $(2,3)$ is structurally zero. Entry $(4,3)$ is a contribution resulting only from elimination of pivot $(1,1)$, and it is needed only when eliminating pivot $(3,3)$. Because of this newly added column, the frontal matrix of node 2 has the minimum structure to carry all the contributions of node 1 to all of its ancestral nodes 2 and 3. The edge between nodes 1 and 3 can be removed, which corresponds to the coupled element 2 absorbing the coupled element 1 in the quotient graph.

We have shown that even if local symmetrization may result in extra fill-ins, it does not symmetrize the adjacency lists of the pivot; it builds at each elimination step the minimal nonsymmetric structure capable of absorbing all the nonsymmetric contributions from all the elements adjacent to the pivot. This nonsymmetric structure is called the [illegible], similar to the symmetric case. By doing so, node $p$ becomes the [illegible] parent of all the nodes $e$ such that $c_e \in \mathcal{R}_p$ or $r_e \in \mathcal{C}_p$ in a tree rooted with the last pivot. The `DMLS` algorithm thus explicitly builds an elimination tree in which each node corresponds to the processing of a nonsymmetric frontal matrix whose structure is defined by $\mathcal{L}_p$ and $\mathcal{U}_p$. This elimination tree is identical to the dependency graph that `MA41_UNS` [6] would build if the same ordering were provided. In fact, the `DMLS` ordering is searching for an ordering that provides a good nonsymmetric elimination tree with respect to some local criterion/metric. The `DMLS` ordering also provides a good estimation of the size of the factors and all the working space required during numerical factorization using the `MA41_UNS` approach. This estimation is exact if the diagonal pivots are numerically stable.

**2.4. The DMLS algorithm.** To design the `DMLS` algorithm, we have exploited many algorithmic techniques from the `AMD` approach [1] and have extended them to the nonsymmetric case. The main difficulty is handling local symmetrization during degree calculation. We first explain how to adapt the symmetric algorithms, then describe the modifications needed for local symmetrization, and conclude this section with a description of the metrics used in pivot selection.

Exploiting identical structures in the graph can greatly speed up the degree update at each elimination step. Two coupled variables $i = (r_i, c_i)$ and $j = (r_j, c_j)$ are said to be [illegible] in $\mathcal{G}$ if they have the same row adjacency structure [illegible] the same column adjacency structure in $\mathcal{G}$ (although the row structure may be different from the column structure). Indistinguishable coupled variables can then be merged into a single so-called [illegible]. We use a boldface letter to denote a supervariable. Thus, $\mathbf{i} = (\mathbf{r}_i, \mathbf{c}_i)$, with $\mathbf{r}_i \equiv \{r_i, r_j\}$, and $\mathbf{c}_i \equiv \{c_i, c_j\}$.

For each row supervariable $\mathbf{r}_i$, let $d_{r_i}$ denote its external row degree [1, 17]. Similarly, for each column supervariable $\mathbf{c}_i$, let $d_{c_i}$ denote its external column degree. The external degrees are defined as

$$d_{r_i} = |\mathcal{A}_{i*} \backslash \mathbf{c}_i| + \left| \left( \bigcup_{e \in \mathcal{R}_i \cup \mathcal{C}_i} \mathcal{U}_e \right) \backslash \mathbf{c}_i \right|, \tag{5}$$

$$d_{c_i} = |\mathcal{A}_{*i} \backslash \mathbf{r}_i| + \left| \left( \bigcup_{e \in \mathcal{C}_i \cup \mathcal{R}_i} \mathcal{L}_e \right) \backslash \mathbf{r}_i \right|. \tag{6}$$

Note that we should consider [illegible] the elements in [illegible] $\mathcal{R}_i$ and $\mathcal{C}_i$ contributing to the row degree and column degree. Indeed, because of local symmetrization, when $(r_i, c_i)$ is selected as the pivot at a later step, those elements will contribute to the structure of both $\mathcal{U}_i$ and $\mathcal{L}_i$ (see (3) and (4)). Therefore, we must ensure that the computed degrees are also consistent with the local symmetrization scheme. This does not mean that we symmetrize all the edges in $\bar{\mathcal{E}}$. It means only that our degree evaluation must anticipate what would happen if $(r_i, c_i)$ were selected as the pivot. That is, during the degree calculation of the uneliminated variables, we need to simulate the effect of local symmetrization. The local symmetrization actually takes place only when a variable is selected as the pivot. This has been illustrated in Figure 3.

Following the symmetric `AMD` algorithm [1], we can approximate the true degrees by their upper bounds, $\bar{d}_{r_i}$ and $\bar{d}_{c_i}$, which, at step $k$, can be computed by

$$\bar{d}_{r_i}^{\,k} = \min \begin{cases} n - k, \\ \bar{d}_{r_i}^{\,k-1} + |\mathcal{U}_p \backslash \mathbf{c}_i|, \\ |\mathcal{A}_{i*} \backslash \mathbf{c}_i| + |\mathcal{U}_p \backslash \mathbf{c}_i| + \displaystyle\sum_{e \in \mathcal{R}_i \cup \mathcal{C}_i} |\mathcal{U}_e \backslash \mathcal{U}_p| - \alpha_i, \end{cases} \tag{7}$$

$$\bar{d}_{c_i}^{\,k} = \min \begin{cases} n - k, \\ \bar{d}_{c_i}^{\,k-1} + |\mathcal{L}_p \backslash \mathbf{r}_i|, \\ |\mathcal{A}_{*i} \backslash \mathbf{r}_i| + |\mathcal{L}_p \backslash \mathbf{r}_i| + \displaystyle\sum_{e \in \mathcal{C}_i \cup \mathcal{R}_i} |\mathcal{L}_e \backslash \mathcal{L}_p| - \beta_i. \end{cases} \tag{8}$$

Note that, unlike the symmetric case, two correction terms $\alpha_i$ and $\beta_i$ have been introduced to improve the accuracy of the approximation to the external degree. Let us justify the $\alpha_i$ term in (7). In the nonsymmetric case, it may happen that $\mathbf{c}_i \notin \mathcal{U}_p$, whereas for an accurate prediction of $\bar{d}_{r_i}$ in the context of local symmetrization, we need to pretend that $\mathbf{c}_i \in \mathcal{U}_p$. In this case, $|\mathbf{c}_i|$ was mistakenly counted in every $|\mathcal{U}_e \backslash \mathcal{U}_p|$ for $e \in \mathcal{C}_i$ and should thus be deducted. The total amount that should be deducted is $\alpha_i = |\mathcal{C}_i| \times |\mathbf{c}_i|$; see [4] for details.

Once $\bar{d}_{r_i}$ and $\bar{d}_{c_i}$ are computed, we have many choices of minimization criteria to select the next pivot. Each choice will lead to a different ordering. One set of criteria or metrics is degree-based, which is a direct function of the degrees (e.g., $\text{Min}(\bar{d}_{r_i} \times \bar{d}_{c_i})$, $\text{Min}(\bar{d}_{r_i} + \bar{d}_{c_i})$, $\text{Min}(\text{Min}(\bar{d}_{r_i}, \bar{d}_{c_i}))$, $\text{Min}(\text{Max}(\bar{d}_{r_i}, \bar{d}_{c_i})))$. Another set is deficiency-based, which is based on estimates of the amount of new fill-in generated at each step. We have experimented several variants of the approximations of the deficiency. Most of the heuristics in [20, 24] can be adapted easily to the nonsymmetric case. Moreover, we have considered a deficiency heuristic that results from discussions with T. Davis and I. S. Duff while working on the approximate minimum degree ordering for symmetric matrices `AMD`. This approximation of the deficiency (referred to as `AMDF` in the symmetric context) is based on the following observation. Suppose $\{r_p, c_p\}$ is the current pivot and the two column elements $e_1$ and $e_2$ are adjacent to $r_i \in \mathcal{L}_p$. In our approximate degree $\bar{d}_{c_i}$ we count twice the row variables that belong to $(\mathcal{L}_{e_1} \backslash \mathcal{L}_p) \cap (\mathcal{L}_{e_2} \backslash \mathcal{L}_p)$. This property can be exploited to improve the estimation of the deficiency, since, in this context, we try to deduct from the degree product the cliques of all the elements adjacent to the current variable. We can consider that $(\mathcal{L}_{e_1} \backslash \mathcal{L}_p) \cap (\mathcal{L}_{e_2} \backslash \mathcal{L}_p) = \emptyset$, because this overlapped term also occurs in the degree product, which is cancelled after subtraction. Thus, for each $r_i \in \mathcal{L}_p$, we can deduct both the area relative to the current clique $p$ (i.e., $|L_p| \times |U_p|$) and the sum of the "external areas" of all the elements adjacent to $(r_i, c_i)$ (i.e., $\sum_{e \in \mathcal{C}_i \cup \mathcal{R}_i} |\mathcal{L}_e \backslash \mathcal{L}_p| \times |\mathcal{U}_p|$). The external area is readily available, since $|\mathcal{L}_e \backslash \mathcal{L}_p|$ has already been computed during the approximate degree calculation. This leads to a more accurate approximation of the deficiency than the approximations introduced in [20, 24] when used in an approximate minimum degree code. This approximation of the deficiency can be easily adapted to our nonsymmetric ordering and will be referred to as `DMLS-MF`. Note that using `AMDF` on symmetric matrices, the amounts of reduction in fill-in and flop count relative to `AMD` have been found to be similar to those reported in [20, 24].

## 3. Numerical experiments.

We now evaluate the `DMLS` ordering algorithm and compare its ordering quality with that obtained by applying both approximate minimum degree and minimum deficiency algorithms on $\mathbf{A} + \mathbf{A}^T$.

### 3.1. Testing environment.

To experiment with our ordering algorithm, we will consider the unsymmetrized multifrontal code `MA41_UNS` [2, 6], which automatically detects and exploits the structural asymmetry of the submatrices involved when processing the elimination tree associated with the pattern of the symmetric matrix $\mathbf{A} + \mathbf{A}^{\mathrm{T}}$. In [7], `MA41_UNS` with `AMD` ordering was shown to be very competitive with `SuperLU` and `UMFPACK` on a large class of matrices including very nonsymmetric ones. We will show in this section that using `DMLS` ordering can significantly improve the speed of `MA41_UNS`. `MA41_UNS` is a tree-based multifrontal algorithm, in which some steps of Gaussian elimination are performed on a dense frontal matrix at each node of the assembly tree, and the Schur complement (or the contribution block) that remains is passed for assembly at the parent node.

`MA41_UNS` can benefit from a numerical scaling of the matrix followed by a numerical preordering (row or column permutations) to maximize the magnitude of the diagonal entries. After numerical pivoting and scaling, a sparsity preserving ordering (symmetric permutation of $\mathbf{A}$) based on an analysis of the pattern of $\mathbf{A} + \mathbf{A}^T$ can be used. The computational graph of the factorization is then computed assuming that diagonal pivots are numerically stable. Since this assumption may not be entirely true during numerical factorization, the solver uses partial pivoting with a threshold value to select numerically stable pivots. It is thus possible that some variables cannot be eliminated from a frontal matrix. The rows and columns containing the noneliminated variables of a frontal matrix are then added to the contribution block and passed to the parent node. Those [illegible] eliminations will result in an increase in the size of the LU factors estimated in the analysis and an increase in the number of operations. In practice, it has been observed that using `MC64` [21, 9, 10] from HSL [15] as preordering can significantly reduce the number of delayed pivots during factorization [3]. This preordering will thus be applied on all our test matrices.

Our test matrices are from the forthcoming Rutherford-Boeing Sparse Matrix Collection [8], the industrial partners of the PARASOL Project,[2] Tim Davis's collection[3], and SPARSEKIT2.[4] Only matrices with structural symmetry less than 0.5 and dimension greater than 1000 were chosen. We define the structural symmetry as the fraction of the nonzeros matched by nonzeros in symmetric locations. Thus, a symmetric matrix has a value of 1, and a highly nonsymmetric matrix has a value close to 0. When there were many similar matrices from the same application domain, we used only a subset with the largest dimensions. Altogether, there were 61 structurally nonsymmetric matrices in our study.

Our computer platform comprises a 2.8 GHz Pentium 4 processor, 2 GBytes of memory, and 1 MByte of cache, with a Linux operating system. We used `gcc -O` to compile the `DMLS` code and `pgf90 -O` to compile all the FORTRAN routines. We also used Goto's BLAS library `libgoto_p4_512-r0.94.so` [14].

We systematically applied random row and column permutations to each matrix. Eleven different permutations were applied to each matrix, and the run that provided the median value of the LU factor size was used in the report.

**3.2. Results.** We first evaluated the quality of the `DMLS` ordering when using different minimization metrics and heuristics mentioned in section 2.4 (min-prod, min-sum, min-min, min-max, and minimum deficiency). Our study showed that `DMLS-MF` (i.e., `DMLS` with approximate minimum deficiency) gives the best quality in terms of fill-in and flop reductions. Therefore, we used `DMLS-MF` in the rest of the experiments. To illustrate the gain in quality we compared `DMLS-MF` with the standard approximate minimum degree algorithm `AMD` as well as `AMDF` (our best local heuristic to approximate the deficiency for the symmetrized matrix $\mathbf{A} + \mathbf{A}^T$).

We observed that for five highly reducible matrices (raefsky5.rua, raefsky6.rua, meg1.rua, bayer05.rua and bayer07.rua) `DMLS-MF` significantly outperformed both `AMD` and `AMDF`—the factor sizes were reduced by 4 to 10 times. Although this is a nice property of `DMLS` it not the scope of our work, since on highly reducible matrices one could consider preprocessing the matrices to first permute them to a block triangular form (BTF) and then search for a symmetric permutation within the diagonal blocks of the BTF format. We have thus excluded these five matrices when reporting the

---

[2] EU ESPRIT IV LTR Project 20160, http://www.parallab.uib.no/projects/parasol.

[3] http://www.cise.ufl.edu/research/sparse/matrices.

[4] http://math.nist.gov/MatrixMarket/data/SPARSKIT.

results, because they will skew the statistics. For the other 56 matrices, we compare in Figure 5 the actual size of the factors (including the extra fill-ins due to numerical pivoting) of the `DMLS-MF`, `AMD`, and `AMDF` orderings. For a relatively large number of matrices (23 with respect to `AMD` and 18 with respect to `AMDF`), the `DMLS-MF` ordering leads to ratios greater than 1.20. Sometimes `DMLS-MF` may give worse ordering than `AMD` or `AMDF`, but it is never less than a ratio of 0.70. Note that there are eight matrices which have structural symmetry less than 0.5 initially but larger than 0.5 after preordering with `MC64`. As expected, for these matrices, relatively smaller gains are obtained from `DMLS`.

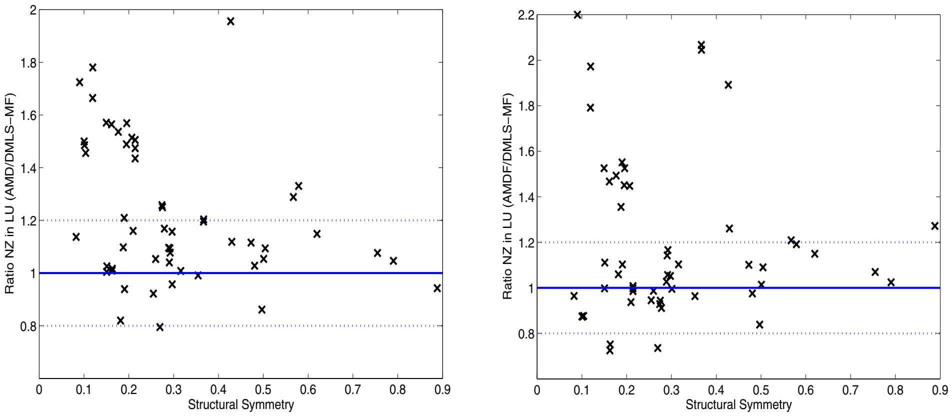


FIG. 5. *Actual fill-in ratios. The x-axis shows the structural symmetry after preprocessing. Left:* `AMD/DMLS-MF`*, mean ratio is* 1.22*, median ratio is* 1.14*; right:* `AMDF/DMLS-MF`*, mean ratio is* 1.20*, median ratio is* 1.6.

In Figure 6, we compare the number of floating-point operations performed during factorization (including numerical pivoting) using the three orderings. For a large number of matrices, the `DMLS-MF` ordering leads to ratios greater than 1.30 for the flop reduction compared to `AMD` (34 matrices) and `AMDF` (23 matrices).

We now focus on 19 large matrices of dimension larger than 10000 and having initial structural symmetry smaller than 0.5 (except for Sandia/mult_dcop_03 and Zhao/Zhao2). This is a subset of the 61 matrices studied above. For this subset, we perform a more detailed quantitative comparison of the `AMDF` and `DMLS-MF` algorithms. These matrices are listed in Table 1 and are sorted in increasing symmetry after the matrices are randomly permuted and reordered using the maximum transversal given by `MC64`. Here, among the 11 symmetry numbers from the 11 initial random permutations, we report the one corresponding to the permutation that gives the mean fill ratio of `AMDF` over `DMLS-MF`.

In Table 2 we report both the estimated factor size given by the analysis phase (columns 2 and 3) and the actual factor size computed during factorization using `MA41_UNS` (columns 5 and 6). Since the pruned frontal matrix structures appeared in factorization are exactly those on which the `DMLS` algorithm is based, the estimation given by `DMLS-MF` is correct modulo small variation due to numerical pivoting. In fact, in addition to an ordering, `DMLS` also gives an assembly tree with the correct frontal size that `MA41_UNS` can use. It is important to note that numerical pivoting has little effect on the structural changes. But this is not the case with `AMDF`, which is based on the graph of the symmetrized matrix $\mathbf{A} + \mathbf{A}^T$. We see that the difference between

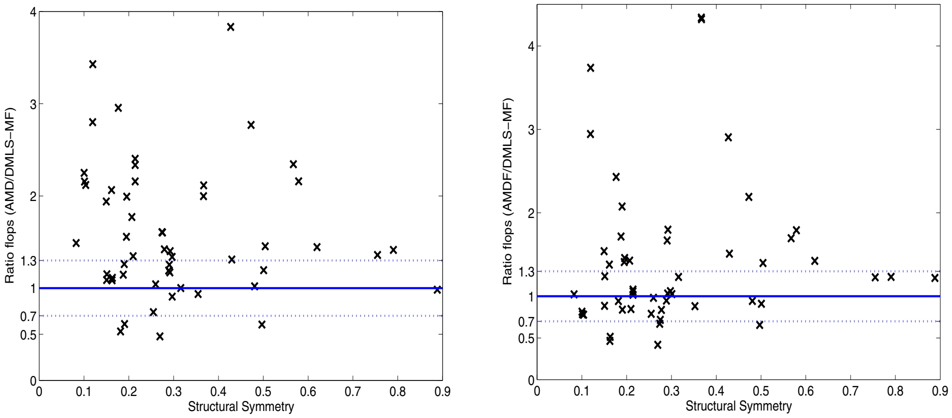


FIG. 6. *Ratio of the number of floating-point operations in factorization. Left:* `AMD/DMLS-MF`*, mean ratio is* 1.64*, median ratio is* 1.39*; right:* `AMDF/DMLS-MF`*, mean ratio is* 1.56*, median ratio is* 1.17*; note that matrix orani*678 *was excluded from the two plots because its flop reduction is almost* 8 *when compared to* `AMDF`*.*

TABLE 1
*Test matrices. StructSym denotes the structural symmetry (both before and after preprocessing).*

| Group/matrix | n | nnz | StructSym Before | StructSym After | Description |
|---|---|---|---|---|---|
| Vavasis/av41092 | 41092 | 1683902 | 0.00 | 0.08 | Unstructured finite element |
| Hollinger/g7jac200sc | 59310 | 837936 | 0.10 | 0.10 | Economic model |
| Hollinger/jan99jac120sc | 41374 | 260202 | 0.00 | 0.16 | Economic model |
| Mallya/lhr34c | 35152 | 764014 | 0.00 | 0.19 | Light hydrocarbon recovery |
| Mallya/lhr71c | 70304 | 1528092 | 0.00 | 0.21 | Light hydrocarbon recovery |
| Hollinger/mark3jac140sc | 64089 | 399735 | 0.22 | 0.21 | Economic model |
| Grund/bayer01 | 57735 | 277774 | 0.00 | 0.25 | Chemical process simulation |
| Hohn/sinc18 | 16428 | 973826 | 0.01 | 0.27 | Single-material crack problem |
| Hohn/sinc15 | 11532 | 568526 | 0.01 | 0.27 | Single-material crack problem |
| Zhao/Zhao2 | 33861 | 166453 | 0.94 | 0.27 | Electromagnetism |
| Hohn/fd18 | 16428 | 63406 | 0.00 | 0.29 | Crack problem |
| Sandia/mult_dcop_03 | 25187 | 193216 | 0.66 | 0.37 | Circuit simulation |
| ATandT/twotone | 120750 | 1224224 | 0.28 | 0.43 | Harmonic balance method |
| ATandT/onetone1 | 36057 | 341088 | 0.10 | 0.43 | Harmonic balance method |
| Norris/torso1 | 116158 | 8516500 | 0.43 | 0.43 | Bioengineering |
| Grund/poli_large | 15575 | 33074 | 0.47 | 0.47 | Chemical process simulation |
| Shen/shermanACb | 18510 | 145149 | 0.26 | 0.50 | Circuit simulation |
| ATandT/pre2 | 659033 | 5959282 | 0.36 | 0.58 | Harmonic balance method |
| Shen/e40r0100 | 17281 | 553562 | 0.33 | 0.89 | Fluid dynamics |

estimation and actual size is significant, and the estimation is often much larger than the actual size. This is because the `MA41_UNS` factorization algorithm can dynamically exploit a more precise frontal matrix structure at each pivot, which can be rectangular and smaller than the frontal matrix structure predicted by `AMDF`. (The frontal matrix predicted by `AMDF` is always square due to initial, global symmetrization $\mathbf{A} + \mathbf{A}^T$.) Furthermore, it has been observed in [6] that an even larger difference can occur in the size of the stack memory. Therefore, after `AMDF` (or `AMD`) ordering and before numerical factorization, one should run a nonsymmetric symbolic factorization algorithm to identify the nonsymmetric structures needed to perform numerical factorization. In our context this extra cost should thus be added to the analysis time when an ordering based on $\mathbf{A} + \mathbf{A}^T$ is used.

In addition to the actual factor size and the floating-point operations, we also report the peak memory (labeled “Real memory” in Table 2) needed to factorize the matrix, which is measured in the number of double precision words. For some classes of matrices (ATandT, Mallya, Norris, Sandia) the `DMLS-MF` ordering leads to much less memory usage than that of `AMDF`. For some other classes of matrices (Grund, Vavasis, Shen), the results are comparable. We found that the Hollinger matrices are very sensitive to the initial random permutations. For example, the number of operations varies between $6.5 \times 10^8$ and $8.3 \times 10^8$ using `AMDF`, between $12.7 \times 10^8$ and $18.4 \times 10^8$ using `DMLS-MF`, and between $17.1 \times 10^8$ and $20.7 \times 10^8$ using `AMD`. Moreover, for this class of matrices, `MA41_UNS` combined with the `AMD` ordering applied to $\mathbf{A} + \mathbf{A}^{\mathrm{T}}$ significantly outperforms all the nonsymmetric solvers considered in [7]. Using `AMDF` thus further reduced the number of operations, and the attempt to exploit the asymmetry of the original matrix did not improve the ordering quality (as shown by the `UMPFPACK` code which attempts to exploit all the asymmetry [7]).

For smaller matrices in the same classes, which are among the complete set of 61 matrices but not shown in Table 2, we have observed a similar behavior. One should point out that on reducible matrices it is always beneficial to first permute to BTF and then apply the ordering to the diagonal block. Furthermore, it has been observed (private communication with Stan Einsenstat) that if one compares the orderings on the largest diagonal block of the BTF, the gains of `DMLS` relative to `AMDF` as reported in this paper are reduced. We feel that this can be only partially explained by the fact that the diagonal blocks of the BTF permuted reducible matrices tend to be structurally more symmetric than the original matrices.

Finally, we report in Table 2 the runtimes of the ordering algorithms. Since both `AMDF` and `DMLS-MF` exploit approximate degree calculations, the complexity of these two codes is directly related to that of the `AMD` ordering. For `DMLS`, since we need to maintain the adjacency structures and the approximate degrees both rowwise and columnwise, we expect `DMLS-MF` to be twice as slow as `AMDF`. This is in general true except for Hohn and Norris classes of matrices, for which `DMLS-MF` is much slower. For Hohn/Sinc* matrices, large dense off-diagonal blocks lead to larger supervariables in the graph of $\mathbf{A} + \mathbf{A}^T$ than in the graph of $\mathbf{A}$. In this case, the asymmetry prevents `DMLS-MF` from selecting larger supervariables, whereas it is not sufficiently nonsymmetric to lead to better ordering. For the matrix Norris/Torso1, the situation is different for at least two reasons. First, taking into account the asymmmetry of the matrix significantly improves the quality of the ordering. Second, it has been shown in our recent work [5] (generalization of the `DMLS` approach to allow off-diagonal and numerical-based pivot selection) that using separate row and column supervariables, one can significantly decrease the ordering time on this class of matrices, and this is true even when pivot selection is restricted to the diagonal as in `DMLS`. However, considering separate row and column supervariables is not at all natural in the `DMLS` context; it would require significant modifications of the data structures used in `DMLS` code and is out of the scope of this work.

In this section, we have focused on the comparison among the local heuristic-based orderings. We believe that improving local heuristics will also benefit the global heuristic orderings that often combine global and local heuristics. Furthermore, we also observed (experiments not reported in this paper) that `DMLS-MF` ordering is at least as good as a nested dissection ordering in preserving sparsity of the factors for most matrices from our set of 19 large matrices.

TABLE 2
*Comparison of* `DMLS-MF` *and* `AMDF` *orderings.*

| Matrix | Size of factors ($10^6$) | | | | | | Flops ($10^9$) | | | Real memory ($10^6$) | | | Ordering time (seconds) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Estimated | | | Actual | | | | | | | | | | | |
| | AMDF | DMLS | Ratio | AMDF | DMLS | Ratio | AMDF | DMLS | Ratio | AMDF | DMLS | Ratio | AMDF | DMLS | Ratio |
| av41092.rua | 11.98 | 9.29 | 1.29 | 9.16 | 9.5 | 0.96 | 3.56 | 3.48 | 1.02 | 9.43 | 9.63 | 0.98 | 0.69 | 2.8 | 0.24 |
| g7jac200sc.rua | 30.62 | 29.84 | 1.03 | 26.12 | 29.87 | 0.87 | 25.25 | 30.98 | 0.82 | 26.94 | 30.88 | 0.87 | 4.75 | 9.31 | 0.51 |
| jan99jac120sc.rua | 4.5 | 4.29 | 1.05 | 3.11 | 4.29 | 0.73 | 0.77 | 1.66 | 0.46 | 3.14 | 4.53 | 0.69 | 1.23 | 2.81 | 0.44 |
| lhr34c.rua | 6.22 | 3.49 | 1.78 | 5.25 | 3.62 | 1.45 | 0.6 | 0.43 | 1.41 | 5.29 | 3.69 | 1.43 | 0.55 | 2.26 | 0.24 |
| lhr71c.rua | 12.77 | 7.21 | 1.77 | 10.75 | 7.43 | 1.45 | 1.27 | 0.89 | 1.43 | 10.77 | 7.49 | 1.44 | 1.24 | 5.31 | 0.23 |
| mark3jac140sc.rua | 19.34 | 14.89 | 1.3 | 14.9 | 14.93 | 1 | 7.68 | 7.26 | 1.06 | 15.26 | 15.41 | 0.99 | 1.38 | 4.23 | 0.33 |
| bayer01.rua | 2.51 | 1.99 | 1.26 | 1.88 | 1.99 | 0.95 | 0.09 | 0.11 | 0.79 | 1.88 | 2 | 0.94 | 0.59 | 1.22 | 0.49 |
| sinc18.rua | 36.76 | 31.65 | 1.16 | 29.28 | 31.72 | 0.92 | 40.84 | 60.22 | 0.68 | 30.11 | 35.39 | 0.85 | 0.98 | 10.84 | 0.09 |
| sinc15.rua | 18.03 | 15.39 | 1.17 | 14.37 | 15.47 | 0.93 | 14.31 | 21.35 | 0.67 | 14.79 | 17.54 | 0.84 | 0.49 | 4.33 | 0.11 |
| Zhao2.rua | 15.79 | 13.9 | 1.14 | 12.97 | 14.23 | 0.91 | 7.69 | 9.19 | 0.84 | 13.78 | 15.04 | 0.92 | 0.45 | 0.95 | 0.47 |
| fd18.rua | 1.44 | 1.05 | 1.38 | 1.1 | 1.07 | 1.03 | 0.11 | 0.12 | 0.95 | 1.11 | 1.12 | 0.99 | 0.1 | 0.18 | 0.53 |
| mult_dcop_03.rua | 2.73 | 0.94 | 2.9 | 1.87 | 0.91 | 2.07 | 0.51 | 0.12 | 4.34 | 1.96 | 0.97 | 2.02 | 3.98 | 0.43 | 9.26 |
| twotone.rua | 22.05 | 8.22 | 2.68 | 15.54 | 8.22 | 1.89 | 14.74 | 5.07 | 2.91 | 15.75 | 9.05 | 1.74 | 1.79 | 2.08 | 0.86 |
| onetone1.rua | 4.85 | 3.2 | 1.51 | 4.04 | 3.2 | 1.26 | 1.94 | 1.28 | 1.51 | 4.09 | 3.65 | 1.12 | 0.33 | 0.51 | 0.64 |
| torso1.rua | 41.29 | 34.08 | 1.21 | 41.28 | 34.19 | 1.21 | 58.69 | 35.91 | 1.63 | 42.34 | 36.62 | 1.16 | 1.7 | 69.42 | 0.02 |
| poli_large.rua | 0.06 | 0.03 | 1.7 | 0.04 | 0.03 | 1.1 | 0 | 0 | 2.19 | 0.04 | 0.03 | 1.11 | 0.02 | 0.01 | 2.16 |
| shermanACb.rua | 0.55 | 0.44 | 1.26 | 0.44 | 0.44 | 1.01 | 0.03 | 0.03 | 0.91 | 0.49 | 0.51 | 0.96 | 2.17 | 0.15 | 14.44 |
| pre2.rua | 115.58 | 90.12 | 1.28 | 107.53 | 90.23 | 1.19 | 405.93 | 226.6 | 1.79 | 161.32 | 99.98 | 1.61 | 17.03 | 61.2 | 0.28 |
| e40r0100.rua | 2.86 | 2.18 | 1.31 | 2.78 | 2.19 | 1.27 | 0.34 | 0.28 | 1.22 | 2.83 | 2.28 | 1.24 | 0.05 | 0.12 | 0.42 |
| Mean | | | 1.48 | | | 1.17 | | | 1.40 | | | 1.15 | | | 1.67 |
| Median | | | 1.29 | | | 1.03 | | | 1.06 | | | 0.99 | | | 0.44 |

## 4. Summary.

In this paper, we have considered the ordering problem for the triangular factorization of a sparse nonsymmetric matrix when pivots can be chosen on the main diagonal. We have described a bipartite quotient graph model for nonsymmetric elimination and have used it as a compact way to represent the elimination graph. The model was first proposed by Pagallo and Maulino [22], but to our knowledge, its implementation did not appear in any literature. Using this model, an ordering algorithm can be implemented in space bounded by the size of the original matrix. This is the so-called in-place property. However, we have found that a straightforward implementation may lead to an algorithm with much higher complexity than an `AMD` type of algorithm applied to the graph of $\mathbf{A} + \mathbf{A}^T$. In order to speed up the ordering algorithm itself, we have introduced the local symmetrization mechanism in the diagonal Markowitz scheme, which allows us to reduce the amount of backtracking needed to update the Schur complement structure at each step. As a result, we have obtained an efficient ordering algorithm both in space and in time—it has the in-place property and the same time complexity as the `AMD` type of algorithms.

We have performed numerical experiments on large numbers of matrices (61) that come from a wide range of applications. The results have showed that our modified diagonal Markowitz scheme indeed can produce better orderings. Compared to the best local greedy algorithms that cannot exploit asymmetry, our algorithm has achieved average gain ratios of 1.22 in factor size and 1.56 in flop count.

**Acknowledgment.** We owe a great deal to the anonymous referees whose careful reading of the initial manuscript and valuable comments have helped us significantly improved the presentation.

## REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[2] P. R. Amestoy and I. S. Duff, *Vectorization of a multiprocessor multifrontal code*, Int. J. Supercomputer Appl., 3 (1989), pp. 41–59.

[3] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and X. S. Li, *Analysis and comparison of two general sparse solvers for distributed memory computers*, ACM Trans. Math. Software, 27 (2001), pp. 388–421.

[4] P. R. Amestoy, X. S. Li, and E. G. Ng, *Diagonal Markowitz Scheme with Local Symmetrization*, Technical report LBNL-53854, Lawrence Berkeley National Laboratory, Berkeley, CA, 2003; also appeared as ENSEEIHT-IRIT report RT/APO/03/05.

[5] P. R. Amestoy, X. S. Li, and S. Pralet, *Unsymmetric ordering using a constrained Markowitz scheme*, SIAM J. Matrix Anal. Appl., to appear.

[6] P. R. Amestoy and C. Puglisi, *An unsymmetrized multifrontal LU factorization*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 553–569.

[7] T. A. Davis, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 165–195.

[8] I. S. Duff, R. G. Grimes, and J. G. Lewis, *The Rutherford-Boeing Sparse Matrix Collection*, Technical report RAL-TR-97-031, Rutherford Appleton Laboratory, Didcot, UK, 1997; also Technical report ISSTECH-97-017 from Boeing Information & Support Services and Report TR/PA/97/36 from CERFACS, Toulouse; http://www.cse.clrc.ac.uk/Activity/SparseMatrices/.

[9] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.

[10] I. S. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.

[11] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford, UK, 1987.

[12] I. S Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325, 1983.

[13] A. GEORGE AND J. W. H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[14] K. GOTO, *High-Performance BLAS*, http://www.cs.utexas.edu/users/flame/goto/.

[15] HSL, *A Collection of Fortran Codes for Large Scale Scientific Computation*, http://www.cse.clrc.ac.uk/Activity/HSL (2000).

[16] X. S. LI AND J. W. DEMMEL, *SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Software, 29 (2003), pp. 110–140.

[17] J. W. H. LIU, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.

[18] J. W. H. LIU, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[19] H. M. MARKOWITZ, *The elimination form of the inverse and its application to linear programming*, Management Sci., 3 (1957), pp. 255–269.

[20] E. G. NG AND P. RAGHAVAN, *Performance of greedy ordering heuristics for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 902–914.

[21] M. OLSHOWKA AND A. NEUMAIER, *A new pivoting strategy for Gaussian elimination*, Linear Algebra Appl., 240 (1996), pp. 131–151.

[22] G. PAGALLO AND C. MAULINO, *A bipartite quotient graph model for unsymmetric matrices*, in Numerical Methods, Lecture Notes in Math. 1005, Springer-Verlag, New York, 1983, pp. 227–239.

[23] D. J. ROSE AND R. E. TARJAN, *Algorithmics aspects of vertex elimination on directed graphs*, SIAM J. Appl. Math., 34 (1978), pp. 176–197.

[24] E. ROTHBERG AND S. C. EISENSTAT, *Node selection strategies for bottom-up sparse matrix ordering*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 682–695.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 245–259

# COMBINATORIAL ANALYSIS OF SINGULAR MATRIX PENCILS*

SATORU IWATA[†] AND RYO SHIMIZU[‡]

**Abstract.** This paper investigates the Kronecker canonical form of matrix pencils under the genericity assumption that the set of nonzero entries is algebraically independent. We provide a combinatorial characterization of the sums of the row/column indices supported by efficient bipartite matching algorithms. We also give a simple alternative proof for a theorem of Poljak on the generic ranks of matrix powers.

**Key words.** Kronecker canonical form, matrix pencil, combinatorial matrix theory, bipartite matching, Dulmage–Mendelsohn decomposition

**AMS subject classifications.** 15A22, 05C50

**DOI.** 10.1137/S0895479804442450

## 1. Introduction.

A matrix pencil is a pair of matrices of the same size. It is often treated as a polynomial matrix whose nonzero entries are of degree at most one. Based on the theory of elementary divisors, Weierstrass established a criterion for strict equivalence, as well as a canonical form, of regular matrix pencils. Somewhat later, Kronecker investigated singular pencils to obtain a canonical form for matrix pencils in general under strict equivalence transformations, which is now called the Kronecker canonical form [12].

The Kronecker canonical form finds a variety of applications in control theory of linear dynamical systems [2, 22, 31, 32]. It is also closely related to the index of differential algebraic equations [13, 14, 29].

Numerically stable algorithms are already available for computing the Kronecker canonical form [1, 3, 8, 19, 20, 35, 36]. Nevertheless, these algorithms are not very accurate in the presence of round-off errors. The numerical difficulty is inherent in the problem as the Kronecker canonical form is highly sensitive to perturbation, which has motivated extensive research on perturbation of matrix pencils [9, 10].

On the other side, matrix pencils arising in applications are often very sparse. It is then desirable to predict the structure of the Kronecker canonical form efficiently from combinatorial information such as the zero/nonzero pattern without numerical computation. Of course, one cannot always obtain exact prediction of the Kronecker canonical form without numerical information. The zero/nonzero pattern, however, determines the indices in the Kronecker canonical form under a genericity assumption that there is no algebraic relation among nonzero entries. One may hope to devise an efficient method for computing them.

---

*Received by the editors March 24, 2004; accepted for publication (in revised form) by B. T. Kågström July 17, 2006; published electronically January 26, 2007. This research is supported in part by a Grant-in-Aid for Scientific Research from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

http://www.siam.org/journals/simax/29-1/44245.html

[†] Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan (iwata@mist.i.u-tokyo.ac.jp).

[‡] Fuji Photo Film Co., Ltd., Asaka, Saitama 351-8585, Japan.

Such a structural approach has been developed for regular matrix pencils by Duff and Gear [5] and Pantelides [29] in the context of differential algebraic equations. Murota [26] described a complete characterization of the Kronecker canonical form of regular matrix pencils in terms of the maximum degree of minors, which is tantamount to the maximum weight of bipartite matchings under the genericity assumption. A recent paper of van der Woude [34] provides another combinatorial characterization based on the Smith normal form [25].

In this paper, we extend the structural approach to the analysis of singular matrix pencils. The possible existence of the minimal row/column indices (rectangular blocks in the canonical form) makes this problem much more complicated than the regular case. In fact, it remains open to design an efficient algorithm for determining the row/column indices. The main result of this paper, however, provides a combinatorial characterization of the sums of the minimal row/column indices using the Dulmage–Mendelsohn decomposition and weighted bipartite matchings. Thus our combinatorial characterization is supported by efficient algorithms.

The structure of the Kronecker canonical form is closely related to the ranks of certain sequences of embedded matrices, called the expanded matrices. We investigate those expanded matrices to show that their ranks are equal to the term-ranks in most cases. As a byproduct, we give a simple alternative proof for a theorem of Poljak [30] on the generic ranks of matrix powers.

The genericity assumption certainly needs some refinement to deal with practical situations. In the description of dynamical systems such as electric circuits, entries that come from conservation laws are exact integers, while other entries that represent physical characteristics of devices are not precise in value because of noises. It should be reasonable to assume the algebraic independence only among the latter types of entries. Based on such an observation, Murota [24, 28] introduced the concept of mixed matrices and investigated their fundamental properties with the aid of matroid theory. In particular, a complete characterization of the Kronecker canonical form of regular mixed matrix pencils is presented in [27]. With a view towards applications, our ultimate target is an extension of this result to singular mixed matrix pencils. Such an investigation, however, requires a solution on generic matrix pencils as the first step.

The outline of this paper is as follows. Section 2 recapitulates the Kronecker canonical form. In section 3, we introduce generic matrix pencils. In section 4, we explain the Dulmage–Mendelsohn decomposition, which plays an essential role in our result on the minimal row/column indices presented in section 5. Sections 6 and 7 are devoted to the analysis of expanded matrices. The simple proof for the theorem of Poljak is shown in section 8. Sections 6 and 8 are independent of sections 4 and 5. Finally, in section 9, we briefly discuss how to use results of combinatorial analysis in the context of numerical computation.

## 2. The Kronecker canonical form of matrix pencils.

Let $D(s) = A + sB$ be an $m \times n$ matrix pencil with the row set $R$ and the column set $C$. We denote by $D(s)[X, Y]$ the submatrix of $D(s)$ determined by $X \subseteq R$ and $Y \subseteq C$. A matrix pencil $D(s) = A + sB$ is said to be regular if $\det D(s) \neq 0$ as a polynomial in $s$. It is strictly regular if both $A$ and $B$ are nonsingular matrices. The rank of $D(s)$ is the maximum size of its submatrix that is a regular matrix pencil. A matrix pencil $\bar{D}(s)$ is said to be strictly equivalent to $D(s)$ if there exists a pair of nonsingular constant matrices $U$ and $V$ such that $\bar{D}(s) = UD(s)V$.

For a positive integer $\mu$, we consider $\mu \times \mu$ matrix pencils $K_\mu$ and $N_\mu$ defined by

$$K_\mu = \begin{pmatrix} s & 1 & 0 & \cdots & 0 \\ 0 & s & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & s & 1 \\ 0 & \cdots & \cdots & 0 & s \end{pmatrix}, \qquad N_\mu = \begin{pmatrix} 1 & s & 0 & \cdots & 0 \\ 0 & 1 & s & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 & s \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

For a positive integer $\varepsilon$, we further denote by $L_\varepsilon$ an $\varepsilon \times (\varepsilon + 1)$ matrix pencil

$$L_\varepsilon = \begin{pmatrix} s & 1 & 0 & \cdots & 0 \\ 0 & s & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & s & 1 \end{pmatrix}.$$

We also denote by $L_\eta^\top$ the transpose matrix of $L_\eta$.

The following theorem establishes the Kronecker canonical form of matrix pencils under strict equivalence transformations. We denote by block-diag$(D_1, \ldots, D_b)$ the block-diagonal matrix pencil with diagonal blocks $D_1, \ldots, D_b$.

THEOREM 2.1 (Kronecker, Weierstrass). [illegible] $D(s)$ [illegible] $U$ [illegible] $V$ [illegible] $\bar{D}(s) = UD(s)V$ [illegible]

$$\bar{D}(s) = \text{block-diag}(H_\nu, K_{\rho_1}, \ldots, K_{\rho_c}, N_{\mu_1}, \ldots, N_{\mu_d}, L_{\varepsilon_1}, \ldots, L_{\varepsilon_p}, L_{\eta_1}^\top, \ldots, L_{\eta_q}^\top, O),$$

[illegible] $\rho_1 \geq \cdots \geq \rho_c > 0$ $\mu_1 \geq \cdots \geq \mu_d > 0$ $\varepsilon_1 \geq \cdots \geq \varepsilon_p > 0$ $\eta_1 \geq \cdots \geq \eta_q > 0$ [illegible] $H_\nu$ [illegible] $\nu$ [illegible] $c$ $d$ $p$ $q$ $\nu$ $\rho_1, \ldots, \rho_c$ $\mu_1, \ldots, \mu_d$ $\varepsilon_1, \ldots, \varepsilon_p$ $\eta_1, \ldots, \eta_q$ [illegible]

The block-diagonal matrix pencil $\bar{D}(s)$ in Theorem 2.1 is often referred to as the Kronecker canonical form of $D(s)$. The numbers $\mu_1, \ldots, \mu_d$ are called the indices of nilpotency. The numbers $\varepsilon_1, \ldots, \varepsilon_p$ and $\eta_1, \ldots, \eta_q$ are the minimal column and row indices, respectively. These numbers together with $\nu, \rho_1, \ldots, \rho_c$ are collectively called the structural indices of $D(s)$.

For a polynomial $g(s)$ in $s$, let $\deg g(s)$ and $\operatorname{ord} g(s)$ denote the highest and lowest degrees of nonvanishing terms of $g(s)$, respectively. By convention, we put $\deg 0 = -\infty$ and $\operatorname{ord} 0 = \infty$. Let $r$ be the rank of $D(s)$. For each $k = 1, \ldots, r$, we denote

$$\delta_k(D) = \max\{\deg \det D(s)[X, Y] \mid |X| = |Y| = k, X \subseteq R, Y \subseteq C\},$$

$$\zeta_k(D) = \min\{\operatorname{ord} \det D(s)[X, Y] \mid |X| = |Y| = k, X \subseteq R, Y \subseteq C\}.$$

The following well-known lemma asserts that $\delta_k$ and $\zeta_k$ are invariant under strict equivalence transformations.

LEMMA 2.2. [illegible] $\bar{D}(s)$ [illegible] $D(s)$ [illegible] $\delta_k(\bar{D}) = \delta_k(D)$ [illegible] $\zeta_k(\bar{D}) = \zeta_k(D)$ [illegible]

[illegible] Since any nonsingular matrix is a product of elementary matrices, it suffices to show that $\delta_k$ and $\zeta_k$ are invariant under strict equivalence transformations by elementary matrices. In particular, we consider a row transformation $\bar{D}(s) = UD(s)$

by an elementary matrix $U$ that is identical with the unit matrix except for the $(v,u)$-component $U_{vu} = \lambda$. This corresponds to adding a multiple of a row $u$ to another row $v$. Let $D(s)[X,Y]$ be an arbitrary submatrix with $|X| = |Y| = k$. Note that $D(s)[X,Y] = \bar{D}(s)[X,Y]$ unless $u \in R \setminus X$ and $v \in X$. If $u \in R \setminus X$ and $v \in X$, then we have

$$\det \bar{D}(s)[X,Y] = \det D(s)[X,Y] + \lambda \det D(s)[X \setminus \{v\} \cup \{v\}, Y],$$

which implies $\deg\det \bar{D}(s)[X,Y] \leq \delta_k(D)$. If, in addition, $\det D(s)[X,Y]$ attains the maximum degree for $\delta_k(D)$, either $\deg\det \bar{D}(s)[X,Y] = \deg\det D(s)[X,Y]$ or $\deg\det \bar{D}(s)[X,Y] < \deg\det D(s)[X,Y] = \deg\det D(s)[X \setminus \{v\} \cup \{u\}, Y]$ holds. Since $\bar{D}(s)[X \setminus \{v\} \cup \{u\}, Y] = D(s)[X \setminus \{v\} \cup \{u\}, Y]$, we have $\deg\det \bar{D}(s)[X \setminus \{v\} \cup \{u\}, Y] = \delta_k(D)$ in the latter case. Thus we obtain $\delta_k(\bar{D}) = \delta_k(D)$.

The same argument applies to an elementary column transformation. Moreover, the invariance of $\zeta_k$ can be shown in a similar manner. □

For the Kronecker canonical form $\bar{D}(s)$ of $D(s)$, we have

$$\zeta_k(D) = \zeta_k(\bar{D}) = \sum_{i=r-k+1}^{c} \rho_i, \qquad \delta_k(D) = \delta_k(\bar{D}) = k - \sum_{i=r-k+1}^{d} \mu_i. \tag{2.1}$$

In particular, $\zeta_k(D) = 0$ for $k = 1, \ldots, r-c$, and $\delta_k(D) = k$ for $k = 1, \ldots, r-d$. Hence, $c = r - \max\{k \mid \zeta_k(D) = 0\}$ and $d = r - \max\{k \mid \delta_k(D) = k\}$ hold. Note that $\delta_k(D)$ is concave in $k$ and $\zeta_k(D)$ is convex in $k$. Moreover, we have $\rho_i = \zeta_{r-i+1}(D) - \zeta_{r-i}(D)$ for $i = 1, \ldots, c$ and $\mu_i = \delta_{r-i}(D) - \delta_{r-i+1}(D) + 1$ for $i = 1, \ldots, d$. Since the sum of the structural indices is equal to the rank of $D(s)$, the equalities in (2.1) for $k = r$ imply that

$$\nu + \sum_{i=1}^{p} \varepsilon_i + \sum_{i=1}^{q} \eta_i = r - \sum_{i=1}^{d} \mu_i - \sum_{i=1}^{c} \rho_i = \delta_r(D) - \zeta_r(D). \tag{2.2}$$

For an $m \times n$ matrix pencil $D(s) = A + sB$, we construct a pair of $km \times kn$ matrices $\Theta_k(D)$ and $\Omega_k(D)$ defined by

$$\Theta_k(D) = \begin{pmatrix} A & O & \cdots & \cdots & O \\ B & A & \ddots & & \vdots \\ O & B & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & A & O \\ O & \cdots & O & B & A \end{pmatrix}, \qquad \Omega_k(D) = \begin{pmatrix} B & O & \cdots & \cdots & O \\ A & B & \ddots & & \vdots \\ O & A & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & B & O \\ O & \cdots & O & A & B \end{pmatrix}.$$

We denote $\theta_k(D) = \operatorname{rank} \Theta_k(D)$ and $\omega_k(D) = \operatorname{rank} \Omega_k(D)$. We also construct a $(k+1)m \times kn$ matrix $\Psi_k(D)$ and a $km \times (k+1)n$ matrix $\Phi_k(D)$ defined by

$$\Psi_k(D) = \begin{pmatrix} A & O & \cdots & O \\ B & A & \ddots & \vdots \\ O & B & \ddots & O \\ \vdots & \ddots & \ddots & A \\ O & \cdots & O & B \end{pmatrix}, \qquad \Phi_k(D) = \begin{pmatrix} A & B & O & \cdots & O \\ O & A & B & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & O \\ O & \cdots & O & A & B \end{pmatrix}.$$

We denote $\psi_k(D) = \operatorname{rank} \Psi_k(D)$ and $\varphi_k(D) = \operatorname{rank} \Phi_k(D)$. Then it is easy to see that the ranks of these expanded matrices are expressed by the structural indices as follows.

THEOREM 2.3. [illegible] $D(s)$ [illegible] $r$ [illegible] $(\nu, \rho_1, \dots, \rho_c, \mu_1, \dots, \mu_d, \varepsilon_1, \dots, \varepsilon_p, \eta_1, \dots, \eta_q)$ [illegible]

$$\theta_k(D) = rk - \sum_{i=1}^{c} \min\{k, \rho_i\}, \qquad \omega_k(D) = rk - \sum_{i=1}^{d} \min\{k, \mu_i\},$$

$$\psi_k(D) = rk + \sum_{i=1}^{p} \min\{k, \varepsilon_i\}, \qquad \varphi_k(D) = rk + \sum_{i=1}^{q} \min\{k, \eta_i\}.$$

[illegible] $\psi_1(D) = r + p$ [illegible] $\varphi_1(D) = r + q$

[illegible] For the matrix pencils $K_\rho$, $N_\mu$, $L_\varepsilon$, $L_\eta^\top$, we have

$$\theta_k(K_\rho) = \mu k - \min\{\rho, k\}, \quad \theta_k(N_\mu) = \mu k, \quad \theta_k(L_\varepsilon) = \varepsilon k, \quad \theta_k(L_\eta^\top) = \eta k,$$

$$\omega_k(K_\rho) = \rho k, \quad \omega_k(N_\mu) = \mu k - \min\{\mu, k\}, \quad \omega_k(L_\varepsilon) = \varepsilon k, \quad \omega_k(L_\eta^\top) = \eta k,$$

$$\psi_k(K_\rho) = \rho k, \quad \psi_k(N_\mu) = \mu k, \quad \psi_k(L_\varepsilon) = \varepsilon k + \min\{\varepsilon, k\}, \quad \psi_k(L_\eta^\top) = \eta k,$$

$$\varphi_k(K_\rho) = \rho k, \quad \varphi_k(K_\mu) = \mu k, \quad \varphi_k(L_\varepsilon) = \varepsilon k, \quad \varphi_k(L_\eta^\top) = \eta k + \min\{\eta, k\}.$$

Summing up these equalities for the blocks in the Kronecker canonical form, we obtain the above formulas. □

COROLLARY 2.4. [illegible] $D(s)$ [illegible] $\Psi_k(D)$ [illegible] $k$ [illegible] $D(s)$ [illegible] $\Phi_k(D)$ [illegible] $k$

[illegible] If $D(s)$ is of column-full rank, the Kronecker canonical form has no minimal column indices. Hence Theorem 2.3 implies $\psi_k(D) = rk$. Similarly, if $D(s)$ is of row-full rank, the Kronecker canonical form has no minimal row indices, which together with Theorem 2.3 implies $\varphi_k(D) = rk$. □

Theorem 2.3 together with (2.1) implies the following corollary.

COROLLARY 2.5. [illegible] $D(s)$ [illegible] $r$ [illegible] $k \geq r$ [illegible] $\theta_k(D) = rk - \zeta_r(D)$ [illegible] $\omega_k(D) = r(k-1) + \delta_r(D)$

[illegible] For $k \geq r$, we have $\theta_k(D) = rk - \sum_{i=1}^{c} \rho_i$ and $\omega_k(D) = rk - \sum_{i=1}^{d} \mu_i$ by Theorem 2.3. Then it follows from (2.1) that $\theta_k(D) = rk - \zeta_k(D)$ and $\omega_k(D) = r(k-1) + \delta_r(D)$ hold for $k \geq r$. □

**3. Generic matrix pencils.** Given a matrix $A$ with the row set $R$ and the column set $C$, we construct a bipartite graph $G(A) = (R, C; E)$ with the vertex sets $R$ and $C$ and the edge set $E$ that consists of nonzero entries of $A$. A subset $M \subseteq E$ is called a matching if no two edges in $M$ share an end-vertex. The term-rank of $A$, denoted by $\operatorname{t-rank} A$, is the maximum size of a matching in $G(A)$. The term-rank provides an upper bound on the rank of $A$. Under the genericity assumption that the set of nonzero entries is algebraically independent, this upper bound is tight. That is, $\operatorname{rank} A = \operatorname{t-rank} A$ holds for a generic matrix. The set function $\tau$ defined by $\tau(X) = \operatorname{t-rank} A[X, C]$ for $X \subseteq R$ is submodular; i.e.,

$$\tau(X) + \tau(Z) \geq \tau(X \cup Z) + \tau(X \cap Z)$$

holds for any $X, Z \subseteq R$. This submodularity will be used in section 7.

A matrix pencil $D(s) = A + sB$ is called a generic matrix pencil if the nonzero entries in $A$ and $B$ are indeterminates (independent parameters). To be more precise, suppose $D(s)$ is a matrix pencil over a field $\mathbf{F}$. That is, $A$ and $B$ are matrices over the field $\mathbf{F}$. Let $\mathbf{K}$ be the prime field of $\mathbf{F}$. A finite set $\mathcal{T} = \{\xi_1, \ldots, \xi_t\} \subseteq \mathbf{F}$ is said to be algebraically independent over $\mathbf{K}$ if there is no nontrivial polynomial $g(x_1, \ldots, x_t)$ over $\mathbf{K}$ such that $g(\xi_1, \ldots, \xi_t) = 0$. A matrix pencil $D(s)$ is generic if the set $\mathcal{T}$ of nonzero entries in $A$ and $B$ is algebraically independent over the prime field $\mathbf{K}$. A typical setting in practice is $\mathbf{F} = \mathbf{R}$ and $\mathbf{K} = \mathbf{Q}$.

For a matrix pencil $D(s) = A + sB$ with the row set $R$ and the column set $C$, let $E$ and $F$ be the sets of edges that correspond to the positions of nonzero entries in $A$ and $B$, respectively. Thus we construct a bipartite graph $G(D) = (R, C; E \cup F)$ with the vertex sets $R$ and $C$ and the edge set $E \cup F$. Since the edges in $E$ and $F$ are distinguished, there may be parallel edges in $G(D)$. Each edge $e$ has a weight $w(e)$ defined by $w(e) = 1$ for $e \in E$ and $w(e) = 0$ for $e \in F$. A subset $M$ of $E \cup F$ is called a matching if no two edges in $M$ share an end-vertex. The maximum size of a matching in $G(D)$ is the term-rank, denoted by $\text{t-rank}\, D(s)$. The weight $w(M) = \sum_{e \in M} w(e)$ of a matching $M$ is equal to the number of edges in $M \cap E$. We denote by $\widehat{\delta}_k(D)$ the maximum weight of a matching of size $k$. We also denote by $\widehat{\zeta}_k(D)$ the minimum weight of a matching of size $k$.

The following two lemmas demonstrate that fundamental quantities of a generic matrix pencil coincide with their combinatorial counterparts; see [28, Theorem 6.2.2] for the proof of Lemma 3.2. It should be emphasized here that these combinatorial counterparts are easy to compute with efficient combinatorial algorithms for bipartite matchings [11, 15, 16, 21, 23, 33].

Lemma 3.1. [illegible] $D(s)$ [illegible] $\operatorname{rank} D(s) = \text{t-rank}\, D(s)$

Lemma 3.2. [illegible] $D(s)$ [illegible] $\delta_k(D) = \widehat{\delta}_k(D)$ [illegible] $\zeta_k(D) = \widehat{\zeta}_k(D)$

It should be remarked that the above definition of genericity is different from that in the previous works [9, 10], where the Kronecker canonical form of a "generic" matrix pencil is known to have a very simple structure with at most two types of blocks [9, section 3.3].

**4. Dulmage–Mendelsohn decomposition.** In this section, we recapitulate the Dulmage–Mendelsohn decomposition of bipartite graphs [6, 7, 8] following the exposition in [28, section 2.2.3].

For a generic matrix pencil $D(s)$ with the row set $R$ and the column set $C$, let $(R_0; R_1; R_\infty)$ and $(C_0; C_1; C_\infty)$ be partitions of $R$ and $C$ such that $|R_0| < |C_0|$ (unless $R_0 = C_0 = \emptyset$), $|R_1| = |C_1|$, and $|R_\infty| > |C_\infty|$ (unless $R_\infty = C_\infty = \emptyset$). Then $D(s)$ is said to be in a block-triangular form with respect to these partitions if it satisfies $D(s)[R_1, C_0] = O$, $D(s)[R_\infty, C_0] = O$, and $D(s)[R_\infty, C_1] = O$. If, in addition, $\operatorname{rank} D(s)[R_j, C_j] = \min(|R_j|, |C_j|)$ holds for $j = 0, 1, \infty$, then $D(s)$ is in a proper block-triangular form. The Dulmage–Mendelsohn decomposition (DM-decomposition) is such a pair of partitions with largest $|R_1| = |C_1|$ that gives a proper block-triangular form. The existence and uniqueness of the DM-decomposition can be verified through the construction below.

Let $\Gamma(Y) \subseteq R$ denote the set of vertices adjacent to $Y \subseteq C$ in $G(D)$. Then the function $f$ defined by

$$f(Y) = |\Gamma(Y)| - |Y| \qquad (Y \subseteq C)$$

is submodular, i.e.,

$$f(Y) + f(Z) \geq f(Y \cup Z) + f(Y \cap Z)$$

holds for any $Y, Z \subseteq C$. The term-rank of the matrix pencil $D(s)$ is characterized by the minimum value of this submodular function $f$, i.e.,

$$\text{t-rank}\, D(s) = \min\{f(Y) \mid Y \subseteq C\} + |C|,$$

which follows from the Hall–Ore theorem for bipartite graphs. Note that the set of minimizers of a submodular function forms a distributive lattice.

Let $Y_0$ denote the unique minimal minimizer of $f$, and $Y_1$ the unique maximal minimizer of $f$. We put

$$\begin{aligned} C_0 &= Y_0, & R_0 &= \Gamma(Y_0), \\ C_1 &= Y_1 \setminus Y_0, & R_1 &= \Gamma(Y_1) \setminus \Gamma(Y_0), \\ C_\infty &= C \setminus Y_1, & R_\infty &= R \setminus \Gamma(Y_1). \end{aligned}$$

Since $f(Y_0) \leq f(\emptyset) = 0$, we have $|R_0| \leq |C_0|$, where the equality holds only if $C_0 = \emptyset$. It follows from $f(Y_0) = f(Y_1)$ that $|R_1| = |C_1|$. Moreover, $f(Y_1) \leq f(C)$ implies $|R_\infty| \geq |C_\infty|$, where the equality holds only if $R_\infty = \emptyset$. The resulting partition $(R_0; R_1; R_\infty)$ and $(C_0; C_1; C_\infty)$ provides the DM-decomposition of $D(s)$. We call $D_0(s) = D(s)[R_0, C_0]$ the horizontal tail and denote its rank by $r_0 = |R_0|$. We also call $D_\infty(s) = D(s)[R_\infty, C_\infty]$ the vertical tail and denote its rank by $r_\infty = |C_\infty|$. The DM-decomposition can be computed efficiently with the aid of bipartite matching algorithms.

**5. The Kronecker canonical form via DM-decomposition.** In this section, we investigate the Kronecker canonical form of a generic matrix pencil $D(s)$ via the DM-decomposition.

LEMMA 5.1. [illegible] $D_0$ [illegible] $D(s)$ [illegible] $\psi_k(D) = \psi_k(D_0) + k(r - r_0)$

[illegible]. Recall $r = |R_0| + |C \setminus C_0|$ and $r_0 = |R_0|$. Since $D_*(s) = D(s)[R \setminus R_0, C \setminus C_0]$ is of column-full rank, so is $\Psi_k(D_*)$ by Corollary 2.4, namely, $\psi_k(D_*) = k(r - r_0)$. Then it follows from $D(s)[R \setminus R_0, C_0] = O$ that $\psi_k(D) = \psi_k(D_0) + k(r - r_0)$. □

Lemma 5.1 together with Theorem 2.3 implies that the minimal column indices of $D(s)$ coincide with those of $D_0(s)$. We now investigate the Kronecker canonical form of the horizontal tail $D_0$.

Let $g_0(s)$ be the monic determinantal divisor

$$g_0(s) = \gcd\{\det D(s)[R_0, Y] \mid |Y| = r_0, Y \subseteq C_0\},$$

where gcd designates the greatest common divisor whose leading coefficient is equal to one. The following lemma is a special case of a theorem of Murota [25] (see also [28, Theorem 6.3.8]). We describe its proof here for completeness.

LEMMA 5.2. [illegible] $g_0(s)$ [illegible] $s$

[illegible]. We first claim that $g_0(s)$ belongs to $\mathbf{K}[s]$. For any column $j \in C_0$, there exists a column subset $Y \subseteq C_0 \setminus \{j\}$ such that $|Y| = r_0$ and $\det D(s)[R_0, Y] \neq 0$. Therefore, for any independent parameter $\xi \in \mathcal{T}$, the monic determinantal divisor $g_0(s)$ is free from $\xi$. Thus $g_0(s)$ is a polynomial over $\mathbf{K}$.

We now suppose that $g_0(s)$ is not a monomial in $s$. Let $\overline{\mathbf{K}}$ be the algebraic closure of $\mathbf{K}$. Then $\mathcal{T}$ is algebraically independent over $\overline{\mathbf{K}}$. On the other hand, there exists a root $\sigma \in \overline{\mathbf{K}} \setminus \{0\}$ that satisfies $g_0(\sigma) = 0$. For a regular submatrix $D(s)[R_0, Y]$, we have $\det D(\sigma)[R_0, Y] = 0$. Since $\det D(\sigma)[R_0, Y]$ is a polynomial in $\mathcal{T}$, this contradicts the algebraic independence of $\mathcal{T}$ over $\overline{\mathbf{K}}$. □

The determinantal divisor is invariant under strict equivalence transformations. Hence $g_0(s)$ is equal to the determinantal divisor of the Kronecker canonical form $\bar{D}_0$ of $D_0$. Then Lemma 5.2 implies that $\bar{D}_0$ does not contain a strictly regular block.

THEOREM 5.3. [illegible]

$$\sum_{i=1}^{p} \varepsilon_i = \widehat{\delta}_{r_0}(D_0) - \widehat{\zeta}_{r_0}(D_0), \tag{5.1}$$

[illegible] $\widehat{\delta}_{r_0}(D_0)$ [illegible] $\widehat{\zeta}_{r_0}(D_0)$ [illegible] $r_0$ [illegible] $G(D_0)$ [illegible]

[illegible]. Since the Kronecker canonical form $\bar{D}_0$ of $D_0$ does not contain a strictly regular block or a rectangular block $L_\eta^\top$, it follows from (2.2) and Lemma 3.2 that $\sum_{i=1}^{p} \varepsilon_i = \delta_{r_0}(D_0) - \zeta_{r_0}(D_0) = \widehat{\delta}_{r_0}(D_0) - \widehat{\zeta}_{r_0}(D_0)$. □

A similar argument applied to the vertical tail $D_\infty$ leads to the following results. Lemma 5.4 together with Theorem 2.3 implies that the minimal row indices of $D(s)$ coincide with those of $D_\infty(s)$. Lemma 5.5 shows that the Kronecker canonical form $\bar{D}_\infty(s)$ of $D_\infty(s)$ does not contain a strictly regular block.

LEMMA 5.4. [illegible] $D_\infty$ [illegible] $D(s)$ [illegible] $\varphi_k(D) = \varphi_k(D_\infty) + k(r - r_\infty)$

LEMMA 5.5. [illegible]

$$g_\infty(s) = \gcd\{\det D(s)[X, C_\infty] \mid X \subseteq R_\infty, |X| = r_\infty\}$$

[illegible] $s$

THEOREM 5.6. [illegible]

$$\sum_{i=1}^{q} \eta_i = \widehat{\delta}_{r_\infty}(D_\infty) - \widehat{\zeta}_{r_\infty}(D_\infty), \tag{5.2}$$

[illegible] $\widehat{\delta}_{r_\infty}(D_\infty)$ [illegible] $\widehat{\zeta}_{r_\infty}(D_\infty)$ [illegible] $r_\infty$ [illegible] $G(D_\infty)$ [illegible]

As an immediate consequence of Theorems 5.3 and 5.6, we have the following theorem implied by (2.2).

THEOREM 5.7. [illegible] $\nu$ [illegible] $\bar{D}(s)$ [illegible] $D(s)$ [illegible]

$$\nu = \widehat{\delta}_r(D) - \widehat{\zeta}_r(D) - \widehat{\delta}_{r_0}(D_0) + \widehat{\zeta}_{r_0}(D_0) - \widehat{\delta}_{r_\infty}(D_\infty) + \widehat{\zeta}_{r_\infty}(D_\infty). \tag{5.3}$$

Note that all these right-hand sides of (5.1), (5.2), and (5.3) can be computed efficiently by the DM-decomposition and weighted bipartite matching algorithms. We have thus obtained a useful combinatorial characterization of the sums of the minimal row/column indices as well as the size of the strictly regular block in the Kronecker canonical form.

Among the structural indices of a generic matrix pencil, $\mu_1, \ldots, \mu_d$ and $\rho_1, \ldots, \rho_c$ are known to be efficiently computable by weighted bipartite matching algorithms.

The results in this section enable us to compute $\nu$, $\sum_{i=1}^{p} \varepsilon_i$ and $\sum_{i=1}^{q} \eta_i$ as well. It still remains open to determine the values of the minimal row/column indices. The obtained partial results, however, provide sufficient information to discern if the Kronecker canonical form contains vertical/horizontal rectangular blocks.

**6. Expanded matrices for indices of nilpotency.** We now turn to the ranks of the expanded matrices, which are in close relation to the structural indices as shown in Theorem 2.3, for a generic matrix pencil $D(s) = A + sB$. Even though the set of nonzero entries in $A$ and $B$ are algebraically independent, the expanded matrices are not generic matrices. It will be shown, however, that the ranks of the expanded matrices are equal to their term-ranks in most cases. In this section, we deal with the expanded matrices $\Theta_k(D)$ and $\Omega_k(D)$, which are particularly related to $\mu_1, \dots, \mu_d$ and $\rho_1, \dots, \rho_c$. The other expanded matrices $\Psi_k(D)$ and $\Phi_k(D)$ will be investigated in section 7.

In order to examine the ranks of $\Theta_k(D)$ and $\Omega_k(D)$, we consider the bipartite graphs $G(\Theta_k(D))$ and $G(\Omega_k(D))$ associated with the expanded matrices. It will turn out that these bipartite graphs allow maximum matchings with periodic structures. As a consequence, the ranks of these expanded matrices are equal to their term-ranks denoted by $\widehat{\theta}_k(D)$ and $\widehat{\omega}_k(D)$.

We first investigate $\Theta_k(D)$. Let $\bar{R}$ and $\bar{C}$ be the row set and the column set of $\Theta_k(D)$. Then $\bar{R} = R^1 \cup \cdots \cup R^k$ and $\bar{C} = C^1 \cup \cdots \cup C^k$, where $R^h$ and $C^h$ are the copies of the row set $R$ and the column set $C$ of $D$ for $h = 1, \dots, k$. For vertices $u \in R$ and $v \in C$, we denote by $u^h$ and $v^h$ the corresponding vertices in $R^h$ and $C^h$. The edge set of the bipartite graph $G(\Theta_k(D))$ consists of $\bar{E} = E^1 \cup \cdots \cup E^k$ and $\bar{F} = F^1 \cup \cdots \cup F^{k-1}$, where $E^h$ and $F^h$ are the copies of $E$ and $F$. The edges in $E^h$ connect $R^h$ and $C^h$, whereas the edges in $F^h$ connect $R^h$ and $C^{h+1}$. In other words, $E^h = \{(u^h, v^h) \mid (u, v) \in E\}$ and $F^h = \{(u^h, v^{h+1}) \mid (u, v) \in F\}$.

For a matching $M$ in $G(D)$, let $M^\circ$ be the set of edges $(u^h, v^h)$ with $(u, v) \in E \cap M$ for $h = 1, \dots, k$ and $(u^h, v^{h+1})$ with $(u, v) \in F \cap M$ for $h = 1, \dots, k-1$. Then $M^\circ$ forms a matching in $G(\Theta_k(D))$. A matching in $G(\Theta_k(D))$ is called a periodic matching if it can be represented as $M^\circ$ with a certain matching $M$ in $G(D)$.

We now introduce the weight $w_k$ on the edge set of $G(D)$ by $w_k(e) = k$ for $e \in E$ and $w_k(e) = k - 1$ for $e \in F$. For a matching $M$ in $G(D)$, we consider the weight of $M$ by $w_k(M) = \sum_{e \in M} w_k(e)$.

LEMMA 6.1. [illegible] $M$ [illegible] $w_k(M)$ [illegible] $G(D)$ [illegible] $M^\circ$ [illegible] $G(\Theta_k(D))$

[illegible]. Suppose to the contrary that $M^\circ$ is not a maximum matching in $G(\Theta_k(D))$. Then there exists an augmenting path with respect to $M^\circ$ in $G(\Theta_k(D))$. Let $P^\circ$ be such an augmenting path with a minimum number of edges. The corresponding set of edges in $G(D)$ forms an augmenting path $P$ with respect to $M$ in $G(D)$. Then we analyze the weight of a matching $M' = M \triangle P$, which is the symmetric difference of $M$ and $P$, i.e., $M \triangle P = (M \cup P) \setminus (M \cap P)$. Suppose that the end-vertices of $P^\circ$ are $u^h \in R^h$ and $v^l \in C^l$. If $h < l$, then the weight of $M'$ satisfies $w_k(M') - w_k(M) = (l - h)(k - 1) - k(l - h - 1) = k - l + h$. On the other hand, if $h \geq l$, then we have $w_k(M') - w_k(M) = (h - l + 1)k - (h - l)(k - 1) = k + h - l$. Thus, in either case, we have $w_k(M') = w_k(M) + k + h - l > w_k(M)$, which contradicts the maximality of $w_k(M)$. $\square$

Let $M^\circ$ be a maximum periodic matching in $G(\Theta_k(D))$ corresponding to the maximum-weight matching $M$ in $G(D)$. We denote by $\partial M^\circ$ the set of end-vertices of the edges in $M^\circ$. Consider the submatrix $\Theta_k(D)[X, Y]$ determined by $X = \bar{R} \cap \partial M$

and $Y = \bar{C} \cap \partial M$. Then the expansion of $\det \Theta_k(D)[X, Y]$ contains a nonzero term

$$\prod_{(u,v)\in M\cap E} {A_{uv}}^{k} \prod_{(u,v)\in M\cap F} {B_{uv}}^{k-1},$$

where $A_{uv}$ and $B_{uv}$ denote the $(u, v)$-components of $A$ and $B$. Each $A_{uv}$ appears exactly $k$ times in $\Theta_k(D)$ and $B_{uv}$ appears exactly $k-1$ times in $\Theta_k(D)$. Hence no other matching cancels this term in the expansion. Thus $\Theta_k(D)[X, Y]$ is a nonsingular submatrix of size $|M^\circ|$, which implies $\theta_k(D) = \widehat{\theta}_k(D)$ by Lemma 6.1.

By interchanging the roles of $A$ and $B$, essentially the same argument leads to $\omega_k(D) = \widehat{\omega}_k(D)$. Thus we obtain the following theorem.

THEOREM 6.2. [illegible] $D(s)$ [illegible] $\theta_k(D) = \widehat{\theta}_k(D)$ [illegible] $\omega_k(D) = \widehat{\omega}_k(D)$

**7. Expanded matrices for column/row indices.** This section is devoted to a combinatorial analysis of the ranks of the expanded matrices $\Psi_k(D)$ and $\Phi_k(D)$ for a generic matrix pencil $D(s)$. Let $\widehat{\psi}_k(D)$ and $\widehat{\varphi}_k(D)$ denote the term-ranks of these matrices.

For $k = 1$, the expanded matrices $\Psi_1(D)$ and $\Phi_1(D)$ are generic matrices and hence their ranks are equal to the term-ranks. This together with Theorem 2.3 enables us to compute $p$ and $q$, the numbers of horizontal and vertical blocks in the Kronecker canonical form, by efficient bipartite matching algorithms.

For general $k$, however, it is not immediately clear that the ranks are equal to the term-ranks. This is because the expanded matrices admit the same entries to appear in different places. In fact, the ranks of the expanded matrices may be less than their term-ranks. For instance, consider a generic matrix pencil

$$D(s) = A + sB = \begin{pmatrix} \alpha_1 & s\beta_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_2 & s\beta_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s\beta_3 & \alpha_3 & 0 & 0 & 0 \\ 0 & 0 & \alpha_4 & 0 & s\beta_4 & \alpha_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_6 + s\beta_5 & s\beta_6 & \alpha_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_8 & s\beta_7 \end{pmatrix}.$$

The Kronecker canonical form of $D(s)$ is $\bar{D}(s) = \text{block-diag}(L_4, L_2)$, which implies

$$\psi_k(D) = \begin{cases} 8k & (k \le 2), \\ 7k + 2 & (2 \le k \le 4), \\ 6k + 6 & (k \ge 4), \end{cases}$$

whereas

$$\widehat{\psi}_k(D) = \begin{cases} 8k & (k \le 3), \\ 6k + 6 & (k \ge 3). \end{cases}$$

Thus $\psi_k(D) = \widehat{\psi}_k(D)$ holds for $k \ne 3$. For $k = 3$, however, we have $\psi_3(D) = 23$ and $\widehat{\psi}_3(D) = 24$.

For $k = 2$, in contrast, the ranks of the expanded matrices coincide with their term-ranks.

THEOREM 7.1. [illegible] $D(s)$ [illegible] $\psi_2(D) = \widehat{\psi}_2(D)$ [illegible] $\varphi_2(D) = \widehat{\varphi}_2(D)$

[illegible]. Let $M^*$ be a maximum matching in $G(\Psi_2(D))$. The row set $\bar{R}$ and the column set $\bar{C}$ are given by $\bar{R} = R_1 \cup R_2 \cup R_3$ and $\bar{C} = C_1 \cup C_2$, where $R_h$ and $C_h$ are copies of $R$ and $C$. For $u \in R$ and $v \in C$, we denote their copies by $u_h \in R_h$ and $v_h \in C_h$. We also denote $X = \bar{R} \cap \partial M^*$ and $Y = \bar{C} \cap \partial M^*$. Then it suffices to show that $W = \Psi_2(D)[X, Y]$ is nonsingular.

Let $M_1^*$ and $M_2^*$ be the sets of edges in $M^*$ incident to $C_1$ and $C_2$, respectively. We denote $X_1 = X \cap \partial M_1^*$, $X_2 = X \cap \partial M_2^*$, $Y_1 = C_1 \cap \partial M^*$, and $Y_2 = C_2 \cap \partial M^*$.

Let $\mathcal{P}$ denote the family of perfect matchings in $G(W)$. For each matching $M \in \mathcal{P}$, we denote $\pi(M) = \prod_{(u,v)\in M} W_{uv}$, where $W_{uv}$ is the $(u, v)$-component of $W$. Recall that

$$\det W = \sum_{M\in\mathcal{P}} \sigma_M \pi(M),$$

where $\sigma_M$ takes 1 or $-1$. We also denote by $\mathcal{P}^\bullet$ the family of perfect matchings $M = M_1 \cup M_2$ such that $\partial M_1 = \partial M_1^*$ and $\partial M_2 = \partial M_2^*$.

We now claim that $\pi(M^\bullet) \neq \pi(M')$ for any pair of $M^\bullet \in \mathcal{P}^\bullet$ and $M' \in \mathcal{P} \setminus \mathcal{P}^\bullet$. Suppose to the contrary that $\pi(M^\bullet) = \pi(M')$. The matching $M'$ contains an edge $(u_2, v_2)$ with $u_2 \in X_1$. Then we have $(u_1, v_1) \in M^\bullet \setminus M'$, and hence $(u_1, z_1) \in M'$ for some $z \in C \setminus \{v\}$, which implies $(u_2, z_2) \in M^\bullet$. This is a contradiction to $u_2 \in X_1$.

Since both $W[X_1, Y_1]$ and $W[X_2, Y_2]$ are generic matrices, we have

$$\sum_{M\in\mathcal{P}^\bullet} \sigma_M \pi(M) = \det W[X_1, Y_1] \cdot \det W[X_2, Y_2] \neq 0.$$

Then the claim above implies $\det W \neq 0$, which means $W$ is nonsingular. □

Furthermore, our analysis on the Kronecker canonical form implies that the ranks of the expanded matrices are equal to their term-ranks for sufficiently large $k$ as follows.

THEOREM 7.2. [illegible] $D(s)$ [illegible] $r$ [illegible] $r_0$ [illegible] $D_0$ [illegible] $k \geq r_0$ [illegible]

$$\psi_k(D) = \widehat{\psi}_k(D) = kr + \widehat{\delta}_{r_0}(D_0) - \widehat{\zeta}_{r_0}(D_0).$$

[illegible]. Due to the submodularity of the term-rank, we have

$$\widehat{\psi}_k(D_0) + \widehat{\varphi}_{k-1}(D_0) \leq \widehat{\theta}_k(D_0) + \widehat{\omega}_k(D_0).$$

Since $D_0$ is of row-full rank, Corollary 2.4 implies $\widehat{\varphi}_{k-1}(D_0) = (k-1)r_0$. It follows from Corollary 2.5 and Theorem 6.2 that $\widehat{\theta}_k(D_0) = \theta_k(D_0) = r_0(k-1) - \delta_{r_0}(D_0)$ and $\widehat{\omega}_k(D_0) = \omega_k(D_0) = r_0 k - \zeta_{r_0}(D_0)$ hold for $k \geq r_0$ Thus, we obtain

$$\widehat{\psi}_k(D_0) \leq kr_0 + \delta_{r_0}(D_0) - \zeta_{r_0}(D_0).$$

Since $\widehat{\psi}_k(D) \leq \widehat{\psi}_k(D_0) + k(r - r_0)$, we have

$$\widehat{\psi}_k(D) \leq kr + \widehat{\delta}_{r_0}(D_0) - \widehat{\zeta}_{r_0}(D_0).$$

On the other hand, Theorems 2.3 and 5.3 imply $\psi_k(D) = rk + \widehat{\delta}_{r_0}(D_0) - \widehat{\zeta}_{r_0}(D_0)$ for $k \geq r_0$. Since $\psi_k(D_0) \leq \widehat{\psi}_k(D_0)$, we have $\psi_k(D) = \widehat{\psi}_k(D) = kr + \widehat{\delta}_{r_0}(D_0) - \widehat{\zeta}_{r_0}(D_0)$ for $k \geq r_0$. □

A similar argument applied to the vertical tail $D_\infty$ leads to the following theorem.

THEOREM 7.3. [illegible] $D(s)$ [illegible] $r$ [illegible] $r_\infty$ [illegible] $D_\infty$ [illegible] $k \geq r_\infty$ [illegible]

$$\varphi_k(D) = \widehat{\varphi}_k(D) = kr + \widehat{\delta}_{r_\infty}(D_\infty) - \widehat{\zeta}_{r_\infty}(D_\infty).$$

**8. Generic matrix powers.** As a byproduct of our combinatorial analysis in section 6, we give a simple alternative proof for a theorem of Poljak [30] on the ranks of powers of generic square matrices.

Let $A$ be an $n \times n$ generic matrix. We associate a directed graph $\vec{G}(A) = (R, \vec{E})$ with the vertex set $R$ identical with the row/column set of $A$. The arc set $\vec{E}$ is the set of nonzero entries of $A$, namely $\vec{E} = \{(u, v) \mid A_{uv} \neq 0\}$. A $k$-walk in $\vec{G}(A)$ is an alternating sequence $(v_0, e_1, v_1, \ldots, e_k, v_k)$ of vertices $v_h \in R$ and $e_h \in \vec{E}$ such that $e_h = (v_{h-1}, v_h)$ for $h = 1, \ldots, k$. A pair of $k$-walks $(v_0, e_1, v_1, \ldots, e_k, v_k)$ and $(v_0', e_1', v_1', \ldots, e_k', v_k')$ is called independent if $v_h \neq v_h'$ holds for $h = 0, 1, \ldots, k$. The following theorem characterizes the rank of $A^k$ in terms of independent $k$-walks.

THEOREM 8.1 (Poljak [30]). [illegible] $A$ [illegible] $A^k$ [illegible] $k$ [illegible] $\vec{G}(A)$

Consider a regular matrix pencil $D(s) = A + sI$, where $I$ denotes the unit matrix. Then a $k$-walk naturally corresponds to a path $P$ in $G(\Theta_k(D))$ from $C^1$ to $R^k$. To be more specific, the path $P$ is given by

$$P = \{({v_{h-1}}^h, {\bar{v_h}}^h) \mid h = 1, \ldots, k\} \cup \{({v_h}^h, {\bar{v_h}}^h) \mid h = 1, \ldots, k\},$$

where $\bar{v_h}$ denotes the column that is identical to the row $v_h \in R$. Then a pair of independent $k$-walks correspond to a pair of vertex-disjoint paths in $G(\Theta_k(D))$. Let $\bar{P} = P_1 \cup \cdots \cup P_\ell$ denote the edge set of $\ell$ such vertex-disjoint paths that come from $\ell$ independent $k$-walks. Then the symmetric difference $\bar{P} \triangle \bar{F}$ forms a matching of size $\ell + n(k-1)$. Conversely, any periodic matching $M^\circ$ can be obtained in this way from a set of independent $k$ walks. Therefore, Lemma 6.1 implies that the maximum number of independent $k$-walks is equal to $\widehat{\theta}_k(D) - (k-1)n$.

On the other hand, we have

$$\operatorname{rank} A^k = \theta_k(D) - (k-1)n.$$

Therefore, in order to prove Theorem 8.1, it suffices to show that $\theta_k(D) = \widehat{\theta}_k(D)$. Since $D(s)$ is not a generic matrix, we cannot directly apply Theorem 6.2. However, we can use essentially the same argument.

Let $M^\circ$ be a maximum periodic matching in $G(\Theta_k(D))$ that corresponds to a matching $M$ in $G(D)$. Consider the submatrix $\Theta_k(D)[X, Y]$ with $X = \bar{R} \cap \partial M$ and $Y = \bar{C} \cap \partial M$. Then the expansion of $\det \Theta_k(D)[X, Y]$ contains a nonzero term

$$\prod_{(u,v) \in M \cap E} {A_{uv}}^k,$$

where $A_{uv}$ denotes the $(u, v)$-component of $A$. Since each $A_{uv}$ appears exactly $k$ times in $\Theta_k(D)$, no other matching cancels this term in the expansion. Thus $\Theta_k(D)[X, Y]$ is a nonsingular submatrix of size $|M^\circ|$, which implies $\theta_k(D) = \widehat{\theta}_k(D)$ by Lemma 6.1.

**9. Discussions.** This paper has investigated the Kronecker canonical form of generic matrix pencils. Even if the genericity assumption is not valid, we can efficiently compute the combinatorial estimates of the sums of the minimal row/column indices as well as the size of the strictly regular block. These estimates may differ from the exact values. However, we can use them for checking if the result of a numerical computation is consistent with the combinatorial information.

For instance, consider a real matrix pencil

$$D(s) = A + sB = \begin{pmatrix} 1 & s & 0 & 30 & 0 & 0 \\ 60 & 0 & s\beta & 0 & 1 & s \\ 0 & 5 & 0 & s & 0 & 0 \\ 0 & 0 & 1 & 0 & s & 1 \\ 0 & 0 & 0 & s & 0 & 0 \end{pmatrix}$$

with a parameter $\beta$. The DM-decomposition provides a pair of permutations of the rows and the columns that transforms $D(s)$ into a block-triangular matrix

$$\widetilde{D}(s) = \begin{pmatrix} s & 1 & 1 & 0 & 0 & 0 \\ 1 & s & s\beta & 60 & 0 & 0 \\ 0 & 0 & 0 & 1 & s & 30 \\ 0 & 0 & 0 & 0 & 5 & s \\ 0 & 0 & 0 & 0 & 0 & s \end{pmatrix}.$$

The Kronecker canonical form of $D(s)$ is $\bar{D}(s) = \text{block-diag}(K_1, N_2, L_2)$ unless $\beta = 1$. If $\beta = 1$, then $\bar{D}(s) = \text{block-diag}(H_2, K_1, N_2, O)$, where $H_2$ is a strictly regular matrix pencil of size 2. This is consistent with the combinatorial estimates obtained as follows.

Let $D'(s)$ be the generic matrix pencil having the same zero/nonzero pattern as $D(s)$. Finding maximum weight bipartite matchings in $G(D')$, we obtain $\delta_1(D') = 1$, $\delta_2(D') = 2$, $\delta_3(D') = 3$, $\delta_4(D') = 4$, and $\delta_5(D') = 3$, which imply that the Kronecker canonical form of $D'(s)$ contains $N_2$. Similarly, we obtain $\zeta_1(D') = 0$, $\zeta_2(D') = 0$, $\zeta_3(D') = 0$, $\zeta_4(D') = 0$, and $\zeta_5(D') = 1$, which imply that the Kronecker canonical form contains $K_1$. The horizontal tail $D'_0$ is of size $2 \times 3$, which means $r_0 = 2$. Furthermore, we have $\delta_2(D'_0) = 2$ and $\zeta_2(D'_0) = 0$. It follows from Theorem 5.3 that the sum of the column indices is equal to 2. Since the DM-decomposition of $D(s)$ does not have a vertical tail, Theorem 5.7 implies $\nu = 0$. Therefore, the Kronecker canonical form of $D'(s)$ is block-diag$(K_1, N_2, L_2)$, which coincides with that of $D(s)$ unless $\beta = 1$.

On the other hand, computer software `guptri` [3, 4] that implements a staircase algorithm may fail to return the correct answer in its default setting of the deflation tolerance `EPSU` $= 10^{-8}$. For example, if we assign $\beta = 40$, then `guptri` returns block-diag$(K_2, N_2, L_1)$ as the Kronecker canonical form. If we assign $\beta = 4$, then the output is block-diag$(K_2, N_1, L_2)$, which is still different from the solution. These phenomena reflect computational difficulty inherent in the problem. In the latter case, however, we are able to detect an error with the aid of combinatorial estimates. In fact, if the Kronecker canonical form is block-diag$(K_2, N_1, L_2)$, we must have $\delta_5(D) = 4$, which contradicts $\delta_5(D) \leq \delta_5(D')$. Then one can obtain the correct answer by trying smaller `EPSU`. Thus, combinatorial analysis may help us to make numerical solutions more reliable.

Another way to use the combinatorial estimates is to design a numerical algorithm that exploits the combinatorial information. If one had an easier way to check the correctness of the estimates, it would lead to a new algorithm particularly efficient for sparse matrices. In fact, such algorithms of combinatorial relaxation type have been developed for the maximum degree of subdeterminants [17, 18, 26, 28]. It would be interesting to devise the same type of algorithms for minimal row/column indices.

**Acknowledgments.** The authors are grateful to Kazuo Murota for fruitful discussions and helpful comments. The authors also thank anonymous referees for many constructive suggestions to improve the presentation.

## REFERENCES

[1] T. Beelen and P. Van Dooren, *An improved algorithm for the computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 105 (1988), pp. 9–65.

[2] J. Demmel and B. Kågström, *Accurate solutions of ill-posed problems in control theory*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 126–145.

[3] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil* $A-\lambda B$*: Robust software with error bounds and applications. Part* I*: Theory and algorithms*, ACM Trans. Math. Software, 19 (1993), pp. 160–174.

[4] J. Demmel and B. Kågström, *The generalized Schur decomposition of an arbitrary pencil* $A - \lambda B$*: Robust software with error bounds and applications. Part* II*: Software and applications*, ACM Trans. Math. Software, 19 (1993), pp. 175–201.

[5] I. Duff and C. W. Gear, *Computing the structural index*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 594–603.

[6] A. L. Dulmage and N. S. Mendelsohn, *Coverings of bipartite graphs*, Canad. J. Math., 10 (1958), pp. 517–534.

[7] A. L. Dulmage and N. S. Mendelsohn, *A structure theory of bipartite graphs of finite exterior dimension*, Trans. Roy. Soc. Canada, Ser. III, 53 (1959), pp. 1–13.

[8] A. L. Dulmage and N. S. Mendelsohn, *Two algorithms for bipartite graphs*, J. Soc. Indust. Appl. Math., 11 (1963), pp. 183–194.

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

[10] A. Edelman, E. Elmroth, and B. Kågström, *A geometric approach to perturbation theory of matrices and matrix pencils. Part II: A stratification-enhanced staircase algorithm*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 667–699.

[11] J. Edmonds and R. M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, in Combinatorial Optimization—Eureka, You Shrink!, Lecture Notes in Comput. Sci. 2570, Springer-Verlag, Berlin, 2003, pp. 31–33.

[12] F. R. Gantmacher, *The Theory of Matrices*, Chelsea, New York, 1959.

[13] C. W. Gear, *Differential-algebraic equation index transformations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 39–47.

[14] C. W. Gear, *Differential algebraic equations, indices, and integral algebraic equations*, SIAM J. Numer. Anal., 27 (1990), pp. 1527–1534.

[15] J. E. Hopcroft and R. M. Karp, *An* $n^{5/2}$ *algorithm for maximum matchings in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.

[16] M. Iri, *A new method of solving transportation-network problems*, J. Oper. Res. Soc. Japan, 3 (1960), pp. 27–87.

[17] S. Iwata, *Computing the maximum degree of minors in matrix pencils via combinatorial relaxation*, Algorithmica, 36 (2003), pp. 331–341.

[18] S. Iwata, K. Murota, and I. Sakuta, *Primal-dual combinatorial relaxation algorithms for the maximum degree of subdeterminants*, SIAM J. Sci. Comput., 17 (1996), pp. 993–1012.

[19] B. Kågström, *RGSVD—An algorithm for computing the Kronecker structure and reducing subspaces of singular* $A-\lambda B$ *pencils*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 185–211.

[20] V. Kublanovskaya, *AB-algorithm and its modification for the spectral problems of linear pencils of matrices*, Numer. Math., 43 (1984), pp. 329–342.

[21] H. W. Kuhn, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart., 2 (1955), pp. 83–97.

[22] A. S. Morse, *Structural invariants of linear multivariable systems*, SIAM J. Control, 11 (1973), pp. 446–465.

[23] J. Munkres, *Algorithms for the assignment and transportation problems*, J. Soc. Indust. Appl. Math., 5 (1957), pp. 32–38.

[24] K. Murota, *Systems Analysis by Graphs and Matroids: Structural Solvability and Controllability*, Springer-Verlag, Berlin, 1987.

[25] K. Murota, *On the Smith normal form of structured polynomial matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 747–765.

[26] K. Murota, *Combinatorial relaxation algorithm for the maximum degree of subdeterminants: Computing Smith-McMillan form at infinity and structural indices in Kronecker form*, Appl. Algebra Engrg. Comm. Comput., 6 (1995), pp. 251–273.

[27] K. Murota, *On the degree of mixed polynomial matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 196–227.

[28] K. Murota, *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin, 2000.

[29] C. C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 213–231.

[30] S. Poljak, *Maximum rank of powers of a matrix of a given pattern*, Proc. Amer. Math. Soc., 106 (1989), pp. 1137–1144.

[31] H. H. Rosenbrock, *State-Space and Multivariable Theory*, John Wiley, New York, 1970.

[32] J. S. Thorp, *The singular pencil of a linear dynamical system*, Internat. J. Control, 18 (1973), pp. 577–596.

[33] N. Tomizawa, *On some techniques useful for solution of transportation network problems*, Networks, 1 (1971), pp. 173–194.

[34] J. W. van der Woude, *The generic canonical form of a regular structured matrix pencil*, Linear Algebra Appl., 353 (2002), pp. 267–288.

[35] P. Van Dooren, *The computation of Kronecker's canonical form of a singular pencil*, Linear Algebra Appl., 27 (1979), pp. 103–140.

[36] J. H. Wilkinson, *Kronecker's canonical form and the QZ algorithm*, Linear Algebra Appl., 28 (1979), pp. 285–303.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 260–278

# SPECTRAL ANALYSIS OF A PRECONDITIONED ITERATIVE METHOD FOR THE CONVECTION-DIFFUSION EQUATION*

DANIELE BERTACCINI†, GENE H. GOLUB‡, AND STEFANO SERRA-CAPIZZANO§

**Abstract.** The convergence features of a preconditioned algorithm for the convection-diffusion equation based on its diffusion part are considered. Analyses of the distribution of the eigenvalues of the preconditioned matrix in arbitrary dimensions and of the fundamental parameters of convergence are provided, showing the existence of a proper cluster of eigenvalues. The structure of the cluster is not influenced by the discretization. An upper bound on the condition number of the eigenvector matrix under some assumptions is provided as well. The overall cost of the algorithm is $O(n)$, where $n$ is the size of the underlying matrices.

**Key words.** finite differences discretization, preconditioning, multilevel structures, convection-diffusion equation

**AMS subject classifications.** 65F10, 65N22, 15A18, 15A12, 47B65

**DOI.** 10.1137/050627381

## 1. Introduction.

The aim of this work is to study the convergence behavior of a preconditioned algorithm to solve the linear systems generated by the discretization of the convection-diffusion equation

$$-\nu\ \nabla\cdot(a(x)\nabla u)+q(x)\cdot\nabla u = f,\quad x\in\Omega, \tag{1.1}$$

$$u = g,\quad x\in\partial\Omega, \tag{1.2}$$

where $\Omega$ is an open region of $\mathbb{R}^d$ with $a(x)$ a uniformly positive function, $q(x)\in\mathbb{R}^d$ a convective velocity field (the wind), $\nabla=(\frac{\partial}{\partial x_1},\dots,\frac{\partial}{\partial x_d})^T$, and $\nu$ the viscosity (or diffusion) coefficient. We stress that models based on similar equations, whose domains can be of dimension $d>3$, arise, e.g., in finance, where each spatial dimension is related to an asset in a basket.

Discretizing problem (1.1) by using centered or upwinding finite differences on equispaced meshes, we reduce the approximate solution of the above problem to the solution of the linear system

$$Ay=b,$$

where the matrix $A$ is nonsymmetric and positive definite and $n$ is the size of $A$; see section 2.2 for more details. If $\Omega$ coincides with $(0,1)^d$ and the stepsizes are given by $(N_j+1)^{-1}$, $N_j\in\mathbb{N}$, $j=1,\dots,d$, $N=(N_1,\dots,N_d)^T$, then the dimension of $A$ is

---

[*] Received by the editors March 23, 2005; accepted for publication (in revised form) by A. J. Wathen July 31, 2006; published electronically February 9, 2007.
http://www.siam.org/journals/simax/29-1/62738.html

[†] Dipartimento di Matematica, Università di Roma "La Sapienza," P.le A. Moro 2, 00185 Roma, Italy (bertaccini@mat.uniroma1.it). The work of this author was supported in part by MIUR grant 2002014121.

[‡] Department of Computer Science, Stanford University, Gates 2B, Stanford, CA 94305 (golub@stanford.edu). The work of this author was supported in part by DOE grant DE-FC02-01ER41177.

[§] Dipartimento di Fisica e Matematica, Università dell'Insubria, Via Valleggio 11, 22100 Como, Italy (stefano.serrac@uninsubria.it, serra@mail.dm.unipi.it). The work of this author was supported in part by MIUR grants 2002014121 and 2004015437, and by Swedish Science Council grant VR 2002-5532.

$n = N_1 \cdot N_2 \cdots N_d$. In the case when $\Omega \subset (0,1)^d$ is a connected domain formed by a finite union of $d$-dimensional rectangles (e.g., L, T, U-shaped domains, etc.), the discretization of the diffusion part of (1.1) is symmetric and positive definite, and the size $n$ will be approximately equal to $m(\Omega) \cdot N_1 \cdot N_2 \cdots N_d$, with $m(\cdot)$ being the usual Lebesgue measure ($m(\Omega) = 1$ for $\Omega = (0,1)^d$). Therefore, when the number of the mesh points in the domain $\Omega$ is large enough, $A$ is large and sparse.

Let us emphasize the dependence of the matrix $A$ on the parameters $a$ and $q$ in (1.1) by writing $A = A(a,q)$ or $A = A(a,q,\Omega)$, where $\Omega$ is the domain. The preconditioner we consider is defined as

$$P = P(a) := D^{1/2}(a)A(1,0)D^{1/2}(a), \tag{1.3}$$

where $D(a)$ is a suitably scaled main diagonal of $A(a,0)$, and $A(1,0)$ denotes the discrete Laplacian ($a = 1$). Preconditioning with a scaled discrete Laplacian operator for nonself-adjoint and nonseparable elliptic boundary value problems was considered in [12] and [15]. Moreover, in [15] the independence of preconditioned iterations from the mesh was observed. The eigenvalue distribution for the diffusive part of the latter problem was investigated in [23, 26, 25].

In this paper we focus our attention on the case when $q$ is nonzero and $\Omega$ is a connected finite union of $d$-dimensional rectangles (a plurirectangle) so that $A(1,0)$ (and consequently the whole preconditioner $P(a)$) is symmetric and positive definite as proven in [26]. In particular, the authors of [23, 26] found that, if $a(x)$ is positive and regular enough and $q(x) \equiv 0$, then the preconditioned sequence shows a proper eigenvalue clustering at the unity (for the notion of proper eigenvalue and singular value clustering, see Definition 2.2), and we prove here that the same holds true in the complex field for problem (1.1) as well. Moreover, under mild assumptions on the coefficients of the problem, we prove that all the eigenvalues of the preconditioned system belong in a complex rectangle $\{z \in \mathbb{C} : \ \mathrm{Re}(z) \in [c, C], \mathrm{Im}(z) \in [-\hat{c}, \hat{c}]\}$ with $c, C > 0$, $\hat{c} \geq 0$ independent of the dimension $n$. Note that the existence of a proper eigenvalue cluster and the aforementioned localization results in the preconditioned spectrum can be very important for fast convergence of preconditioned iterations (see, e.g., [4]): here we will use and generalize to the case of nonnormal preconditioners a recent general tool devised in [24] for deducing the eigenvalue clustering from the singular value clustering, the latter being much easier to check.

In previous works [1, 5] solvers based on the symmetric/skew-symmetric splittings of $A$ were considered. We stress that symmetric/skew-symmetric splittings can be used successfully as preconditioners; see [2].

Indeed, beside the spectral theoretical analysis of the preconditioned structures, the idea is to propose a technique that can be easily used. In fact, the ingredients are a Krylov method (e.g., GMRES, BiCGSTAB, etc.), a matrix vector routine (for sparse or even diagonal matrices), and a solver for the related diffusion equation with a constant coefficient (a method based, e.g., on the cyclic reduction approach [9, 14] or on multigrid methods [27, 19] for which professional software is available). Of course, if the convection part is dominating, then the considered approach can be enriched by approximating the related discrete operator. We stress that convection-dominated problems require appropriate upwind discretization to avoid spurious oscillations.

### 1.1. Outline.

The paper is organized as follows. In section 2 some tools and definitions from structured linear algebra are introduced, while in section 3 the preconditioner and some of its basic properties are introduced. In sections 4 and 5 we first derive specific tools for dealing with eigenvalue clusters and then we study the spectral properties of the preconditioned matrix sequences, with special emphasis on the eigenvalue and singular value clusterings. Section 6 is devoted to the convergence analysis of GMRES. Moreover, some numerical experiments in both two dimensions and three dimensions, and their computational aspects, are presented and discussed. Section 7 concludes the paper with some final comments and remarks.

## 2. Preliminaries.

We start by stating a few results from the spectral theory of Toeplitz matrix sequences (subsection 2.1) and then we briefly analyze the structure of the coefficient matrix $A$ (subsection 2.2).

### 2.1. Definitions and tools for sequences of Toeplitz matrices.

Let $f$ be a $d$-variate Lebesgue integrable function defined over the hypercube $\mathcal{T}^d$, with $\mathcal{T} = (-\pi, \pi]$ and $d \geq 1$. From the Fourier coefficients of $f$ (called a symbol or generating function)

$$a_j = \frac{1}{(2\pi)^d} \int_{\mathcal{T}^d} f(z) e^{-\mathrm{i}(j,z)}\, dz, \qquad \mathrm{i}^2 = -1, \quad j = (j_1, \ldots, j_d) \in \mathbb{Z}^d, \tag{2.1}$$

with $(j,z) = \sum_{r=1}^{d} j_r z_r$, one can build the sequence of Toeplitz matrices $\{T_N(f)\}_N$, $N = (N_1, \ldots, N_d)$, where $T_N(f) \in \mathbb{C}^{n \times n}$ and $n = \prod_{r=1}^{d} N_r$. The matrix $T_N(f)$ is said to be the Toeplitz matrix of order $N$ generated by $f$ (see, e.g., [8] for more details).

For example, if $d = 1$ we have that $a_j$, $j = -(N_1 - 1), \ldots, 0, \ldots, (N_1 - 1)$, is the value on the $j$th diagonal of the $N_1 \times N_1$ Toeplitz matrix $T_{N_1}$. The Fourier coefficients $a_j$ are equal to zero (for $|j|$ large enough) if $f$ is a (multivariate) trigonometric polynomial. Therefore, the corresponding Toeplitz matrix is multilevel and banded. A typical example is the case of the classical $d$-level Laplacian with Dirichlet boundary conditions, discretized by equispaced finite difference formulas over a square region. For instance, the generating function of the (negative) Laplacian (discretized by centered differences of accuracy order 2 and minimal bandwidth) is expressed by

$$\sum_{j=1}^{d} (2 - 2\cos(z_j)).$$

For $d = 1$ the corresponding matrix is the symmetric tridiagonal matrix $T_{N_1} = \text{Toeplitz}(-1, 2, -1)$ while, in the general case, it corresponds to $\sum_{j=1}^{d} P_j$ with

$$P_j = I_{N_1} \otimes \cdots \otimes I_{N_{j-1}} \otimes T_{N_j} \otimes I_{N_{j+1}} \otimes \cdots \otimes I_{N_d}.$$

The spectral properties of the sequence $\{T_N(f)\}_N$ and of related preconditioned sequences are completely understood and characterized in terms of the underlying generating functions. For instance, $T_N(f) = T_N^*(f)$ ($^*$ is the transpose conjugate operator) for every $N$ if and only if $f$ is real valued: more results are given in Theorem 2.1 following. Before stating it we clarify some notation that we will use throughout the paper.

We consider two nonnegative function $\alpha(\cdot)$ and $\beta(\cdot)$ defined over a domain $D$ with accumulation point $\bar{x}$ (if $D = \mathbb{N}$, then $\bar{x} = \infty$; if $D = \mathcal{T}^d$, then $\bar{x}$ can be any point of $D$). We write

- $\alpha(\cdot) = O(\beta(\cdot))$ if and only if there exists a pure positive constant $K$, such that $\alpha(x) \leq K\beta(x)$, for every (or for almost every) $x \in D$ (here and in the following, by pure or universal constant we mean a quantity not depending on the variable $x \in D$);

- $\alpha(\cdot) = o(\beta(\cdot))$ if and only if $\alpha(\cdot) = O(\beta(\cdot))$ and $\lim_{x\to\bar{x}} \alpha(x)/\beta(x) = 0$ with $\bar{x}$ a given accumulation point of $D$, which will be clear from the context;
- $\alpha(\cdot) \sim \beta(\cdot)$ if and only if $\alpha(\cdot) = O(\beta(\cdot))$ and $\beta(\cdot) = O(\alpha(\cdot))$;
- $\alpha(\cdot) \approx \beta(\cdot)$ if and only if $\alpha(\cdot) \sim \beta(\cdot)$ and $\lim_{x\to\bar{x}} \alpha(x)/\beta(x) = 1$ with $\bar{x}$ a given accumulation point of $D$ (the latter can be rewritten as $\alpha(x) = \beta(x)(1+o(1))$ with $1+o(1)$ uniformly positive in $D$).

Theorem 2.1 (see [8, 22]). [illegible] $f$ [illegible] $g$ [illegible] $d$ [illegible] $\mathcal{T}^d$ [illegible] $g$ [illegible]

1. [illegible] $f$ [illegible] $T_N(f)$ [illegible] $(m, M)$ [illegible] $m =$ [illegible] $f$ [illegible] $M =$ [illegible] $f$.
2. [illegible] $\lambda_{\min}(T_N)$ [illegible] $\lambda_{\max}(T_N)$ [illegible] $T_N(f)$ [illegible]

$$\lim_{N\to\infty} \lambda_{\min}(T_N) = m, \qquad \lim_{N\to\infty} \lambda_{\max}(T_N) = M;$$

3. [illegible] $N_i \sim N_j$ [illegible] $i$ [illegible] $j$ [illegible] $\lambda_{\min}(T_N) - m \sim n^{-\alpha/d}$ [illegible] $M - \lambda_{\max}(T_N) \sim n^{-\beta/d}$ [illegible] $N_i \approx \alpha_{i,j} N_j$ [illegible] $i, j$ [illegible] $\alpha_{i,j}$ [illegible] $\lambda_{\min}(T_N) - m \approx c_m n^{-\alpha/d}$ [illegible] $M - \lambda_{\max}(T_N) \approx c_M n^{-\beta/d}$. [illegible] $\alpha$ [illegible] $f(z) - m$ $\beta$ [illegible] $M - f(z)$ [illegible] $c_m, c_M$ [illegible]

Definition 2.2. [illegible] $\{A_n\}_n$ $A_n$ [illegible] $n$ [illegible] $p \in \mathbb{C}$ [illegible] $\epsilon > 0$ [illegible] $A_n$ [illegible] $D(p,\epsilon) = \{z \in \mathbb{C} : |z-p| < \epsilon\}$ [illegible] $\epsilon$ [illegible] $n$ [illegible] $A_n$ [illegible] $n$ [illegible] $p$ [illegible] $D(p,\epsilon)$ [illegible] $(p-\epsilon, p+\epsilon)$

[illegible] $\{A_n\}_n$ $A_n$ [illegible] $n$ [illegible] $p \in \mathbb{R}_0^+$ [illegible] $\epsilon > 0$ [illegible] $A_n$ [illegible] $(p-\epsilon, p+\epsilon)$ [illegible] $\epsilon$ [illegible] $n$

**2.2. The discrete problem and splitting the contribution of convection and diffusion.** We denote with $\mathrm{Re}(G)$ the symmetric and with $\mathrm{i}\,\mathrm{Im}(G)$ the skew-symmetric part of a real coefficient matrix $G$, i.e., $\mathrm{Re}(G) = (G+G^*)/2$ and $\mathrm{Im}(G) = (G-G^*)/(2\mathrm{i})$, respectively.

The analysis is performed without restrictions on the dimension $d$ of problem (1.1), provided that $a(x) > 0$ and that the domain is a hypercube (by exploiting the analysis in [26], the same can be extended to the case when the domain is a connected finite union of $d$-dimensional rectangles by using the same arguments as in [5]). Conversely, we emphasize that here the numerical tests are performed mainly on two-dimensional problems with $a(x) > 0$.

Note that we can always write

$$A = \Theta(a) + \Psi(q),$$

where the matrix $\Theta(a) = A(a,0)$ is the discretization of the diffusion term, and the matrix $\Psi(q)$ is the discretization of the convection term. We observe that when $q(x) = (w_1, w_2, \ldots, w_d)^T$ is a constant vector and a centered difference discretization is used, the matrix $\Psi(q)$ is skew-symmetric and coincides with the $d$-level Toeplitz structure that, for $d=2$, is given by

$$S_{N_1} \otimes I_{N_2} + I_{N_1} \otimes S_{N_2},$$

where $S_{N_k}$, $k=1,2$, is the Toeplitz matrix generated by $f(z) = (-2\mathrm{i}w_k/(2h_k))\sin(z)$, i.e.,

$$S_{N_k} = \frac{w_k}{2h_k} \begin{pmatrix} 0 & 1 & & & \\ -1 & & & & \\ & \ddots & \ddots & \ddots & \\ & & & & 1 \\ & & & -1 & 0 \end{pmatrix}_{N_k \times N_k} . \tag{2.2}$$

On the other hand, $\Theta(a)$ is a $d$-level Toeplitz matrix which, for $d=2$, is given by

$$T_{N_1} \otimes I_{N_2} + I_{N_1} \otimes T_{N_2},$$

where, if $a(x)=1$, $T_{N_k}$ is the usual one-dimensional discrete Laplacian with generating function given by $(\nu/h_k^2)(2-2\cos(z))$, i.e., the tridiagonal Toeplitz matrix

$$T_{N_k} = \frac{\nu}{h_k^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & & & & \\ & \ddots & \ddots & \ddots & \\ & & & & -1 \\ & & & -1 & 2 \end{pmatrix}_{N_k \times N_k} .$$

For the upwind scheme we consider here, if $q(x)$ is a constant vector, the matrix $A$ is as before with the exception of $S_{N_k}$ as in (2.2), which is now the following bidiagonal matrix:

$$S'_{N_k} = \frac{w_k}{h_k} \begin{pmatrix} 1 & 0 & & & \\ -1 & & & & \\ & \ddots & \ddots & \ddots & \\ & & & & 0 \\ & & & -1 & 1 \end{pmatrix}_{N_k \times N_k} . \tag{2.3}$$

For simplicity, from here on we consider $h_k = h$, $k = 1, \dots, d$, and we normalize the underlying linear systems by multiplying the left and right sides by $h^2$.

As in the case of the upwind scheme considered above, the symmetric part of $A$ cannot be exactly the discretization of the diffusion term $\Theta(a)$, and the skew-symmetric part of $A$ cannot be exactly the discretization of the convection term $\Psi(q)$. Indeed, we observed (see [5, Theorem 3.5, p. 466] and Remark 3.2 in [5]) the following property for a centered difference discretization of (1.1).

THEOREM 2.3. [illegible] $\nabla \cdot q(x)$ [illegible] (1.1) [illegible] (1.1) [illegible] 2 [illegible]

$$\mathrm{Re}(A(a,q)) = \Theta(a) + E,$$

$$\mathrm{iIm}(A(a,q)) = \Psi(q) - E,$$

$$E = \frac{\Psi(q) + \Psi^*(q)}{2} \tag{2.4}$$

$$\|E\|_2 \leq c_d\, h^2 \tag{2.5}$$

$$c_d = \alpha_d \|\nabla \cdot q\|_\infty$$

$\alpha_d = 2d$ $d = 2$ $d = 3$ $\Omega = (0,1)^d$

For the upwind scheme based on (2.3), we have that

$$||E||_2 \leq h\, \alpha_d' \max_{x \in \overline{\Omega}} |q(x)|,$$

where $\alpha_d'$ is a constant of the order of unity which depends only on $d$ and the discretization.

Under the assumption that $\|\nabla \cdot q\|_\infty$ is smaller than a suitable positive constant, by using the same arguments as in [5], we can prove that $\mathrm{Re}(A)$ is real symmetric positive definite but ill-conditioned with a condition number asymptotic to $h^{-2}$.

## 3. The preconditioner.

Here we focus on certain Krylov methods (e.g., GMRES; see [20] and [10]) preconditioned by

$$P := P(a, \Omega) = D^{1/2}(a)\Theta(1)D^{1/2}(a), \quad \Theta(1) = A(1, 0), \tag{3.1}$$

where $D(a)$ is a diagonal matrix which, in MATLAB notation, is given by

$$D(a) = \frac{1}{\gamma}\mathrm{diag}\left(\mathrm{diag}\left(\Theta(a)\right)\right), \qquad \gamma = \Theta(1)_{j,j}.$$

For example, if we consider the centered difference approximation of the Laplacian $\Theta(1)$, we have $\gamma = 4$ for $d = 2$ and $\gamma = 6$ for $d = 3$, where $d$ is the dimension of the domain of the problem. Note that $P$ in (3.1) is an approximation of the matrix generated by the discretization of the diffusive part of (1.1). Similar strategies were used in [11], in [15], and in [23, 25] for the purely diffusive equation, or, in other words, with $q$ as a null vector in (1.1).

The resolution of linear systems with matrices as in (3.1) can be performed within a linear arithmetic cost by means of fast Poisson solvers, and this is important for an efficient implementation of (3.1). Classical (direct) Poisson solvers are mainly based on cyclic reduction or on multigrid algorithms (see [9, 14] and, e.g., [27, 19]). From Theorem 2.3, we infer that $A$ is certainly positive definite if the norm of $E$ is smaller than the minimum (positive) eigenvalue of $\Theta(a)$. More specifically

$$\min_j(\lambda_j(\Theta(a))) \geq \nu h^2\, m(\Omega)\beta_d \min_{\overline{\Omega}} a,$$

with $m(\cdot)$ denoting the Lebesgue measure. Therefore, by using again the bound in Theorem 2.3 and by following the same arguments as in [5, Theorems 3.6 and 3.7], it is easy to prove the following two results, which are important in order to gain insight into the convergence of preconditioned iterations. From here on, where not otherwise stated, we will consider the centered differences discretization of precision order 2 and minimal bandwidth for (1.1).

THEOREM 3.1. [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] (1.1) [illegible]

$$\|\nabla \cdot q\|_\infty < \nu \frac{\beta_d}{\alpha_d} m(\Omega) \min_{\overline{\Omega}} a$$

[illegible] $a(x)$ [illegible] $\mathbf{C}^2(\overline{\Omega})$ [illegible] $\{P^{-1}\mathrm{Re}(A)\}_n$ [illegible] 1 [illegible] $[c, C]$ [illegible]

THEOREM 3.2. [illegible] 3.1 [illegible] $\{P^{-1}\mathrm{Im}(A)\}_n$ [illegible] 0 [illegible] $[-\hat{c}, \hat{c}]$ $\hat{c} > 0$

In reference to Theorem 3.1 we have $\beta_3 = 3\pi^2$ (i.e., for $d = 3$) and $\beta_2 = 2\pi^2$ if centered differences are used in (1.1). Therefore, for $d = 3$, $\Omega = (0,1)^d$ (three dimensions), the hypothesis on $q$ in Theorem 3.1 reads $||\nabla \cdot q||_\infty < \nu\pi^2/2$, which can be quite restrictive. However, if the latter is not satisfied, then everything in Theorems 3.1 and 3.2 can be stated identically, except for the fact that the interval $[c, C]$ (only in Theorem 3.1), with $c, C$ still independent of $n$, may include 0. The same can be stated for the subsequent and more important Theorem 4.3. In conclusion, the eigenvalue spectral clustering is not affected by the considered assumption, while the localization is affected only partially. However, we stress that we experienced the existence of good localization results even with weaker hypotheses than those in Theorems 3.1 and 3.2.

**4. The cluster.** To understand the behavior of preconditioned iterations, we analyze the spectrum of the coefficient matrix associated with (1.1) after preconditioning and the related matrix of eigenvectors; see, e.g., [10, 4]. First, we prove the existence of a proper cluster of the singular values through the decomposition of the preconditioned matrices as identity plus low-norm plus low-rank (Theorem 4.1). Second, we derive a general result (Theorem 4.3) on the relationships between proper eigenvalue and singular value clusters. From the latter result and from Theorems 3.1 and 3.2, we deduce the eigenvalue uniform boundedness and proper eigenvalue clustering in Corollary 4.4 and, in section 5, we provide some inequalities for the eigenvalues. Finally, we give a bound for the condition number of the matrix of the eigenvectors and discuss the convergence of GMRES in section 6.

THEOREM 4.1. [illegible] 3.1 [illegible] $\epsilon > 0$ [illegible] $\bar{N} = (\bar{N}_1, \ldots, \bar{N}_d)$ ([illegible] $\mathbb{N}^d$) $\bar{N} = \bar{N}(\epsilon)$ $r = r(\epsilon) < n$ [illegible]

$$N = (N_1, \ldots, N_d) > \bar{N}(\epsilon) = (\bar{N}_1, \ldots, \bar{N}_d),$$

[illegible]

$$P^{-1/2}AP^{-1/2} = I + R^{(1)} + R^{(2)}, \tag{4.1}$$

[illegible] $||R^{(1)}||_2 \leq \epsilon$ [illegible] $\mathrm{rank}(R^{(2)}) \leq r$ [illegible] $\{P^{-1/2}AP^{-1/2}\}_n$ [illegible] 1 [illegible]. We can write

$$P^{-1/2}AP^{-1/2} = P^{-1/2}(\mathrm{Re}(A) + \mathrm{i}\mathrm{Im}(A))P^{-1/2}$$
$$= P^{-1/2}(\Theta(a) + E)P^{-1/2} + P^{-1/2}(\Psi(q) - E)P^{-1/2},$$

and, from Theorem 3.1, we have that, with fixed $\epsilon_1 > 0$ small enough, there exist $\tilde{N} = (\tilde{N}_1, \ldots, \tilde{N}_d)$ and a constant $r_1$ such that for $N > \tilde{N}$ (to be intended componentwise), $n - r_1$ eigenvalues of the matrix $P^{-1/2}\mathrm{Re}(A)P^{-1/2}$ belong to the interval $(1-\epsilon_1, 1+\epsilon_1)$, and all the eigenvalues of $P^{-1/2}\mathrm{Re}(A)P^{-1/2}$ belong to an interval $[c, C]$, $0 < c < C$; i.e., we can write

$$P^{-1/2}\mathrm{Re}(A)P^{-1/2} = I + R_1^{(1)} + R_1^{(2)}, \tag{4.2}$$

where $||R_1^{(1)}||_2 \le \epsilon_1$ and $\mathrm{rank}(R_1^{(2)}) \le r_1$.

Moreover, from Theorem 3.2, we infer that the matrix sequence

$$\{P^{-1/2}\mathrm{Im}(A)P^{-1/2}\}_n$$

is spectrally bounded and clustered at zero; i.e., for $N$ large enough,

$$\mathrm{i}P^{-1/2}\mathrm{Im}(A)P^{-1/2}$$

is a skew-symmetric matrix whose eigenvalues are in $[-\mathrm{i}\hat{c}, \mathrm{i}\hat{c}]$. Therefore, there exist $\hat{N} = (\hat{N}_1, \ldots, \hat{N}_d)$ and a constant $r_2$ such that for $N > \hat{N}$, $n - r_2$ eigenvalues of $P^{-1/2}\mathrm{Im}(A)P^{-1/2}$ belong to $(-\epsilon_2, \epsilon_2)$ and all the eigenvalues of $P^{-1/2}\mathrm{Im}(A)P^{-1/2}$ belong to $[-\hat{c}, \hat{c}]$. Then, we can write

$$P^{-1/2}\mathrm{Im}(A)P^{-1/2} = R_2^{(1)} + R_2^{(2)}, \tag{4.3}$$

where $||R_2^{(1)}||_2 \le \epsilon_2$ and $\mathrm{rank}(R_2^{(2)}) \le r_2$, $||P^{-1/2}\mathrm{Im}(A)P^{-1/2}||_2 = \hat{c}$. The claimed results follow by taking

$$R^{(1)} = R_1^{(1)} + R_2^{(1)}, \quad \epsilon = \epsilon_1 + \epsilon_2; \quad r = r_1 + r_2, \quad \bar{N} = \max\{\hat{N}, \tilde{N}\}, \tag{4.4}$$

where the condition for $\bar{N}$ is to be intended componentwise. Finally, the existence of a proper singular value cluster at 1 of the sequence $\{P^{-1/2}AP^{-1/2}\}_n$ is a direct consequence of (4.1) and of the singular value decomposition [17]. □

Note that $r$ in (4.4) does not depend on $N$ for $N > \overline{N}$ because of the existence of a proper cluster for the spectrum of

$$\{P^{-1/2}\mathrm{Re}(A)P^{-1/2}\}_n$$

and of

$$\{P^{-1/2}\mathrm{Im}(A)P^{-1/2}\}_n.$$

Now we introduce a general tool, i.e., Theorem 4.3, for analyzing eigenvalue clusters of a preconditioned matrix sequence. We will take recourse to the following result (Theorem 4.2) essentially based on the majorization theory (see, e.g., [7]).

Theorem 4.2 (see [24]). [illegible] $\{A_n\}_n$ [illegible] $n$ [illegible] $\{A_n\}_n$ [illegible]

Theorem 4.3. [illegible] $\{A_n\}_n$ [illegible] $\{P_n\}_n$ [illegible] $P_n$ [illegible] $B_n$ $C_n$ [illegible] $U_n$ [illegible] $U_n$ [illegible] $A_n = B_n + C_n$ [illegible]

1. [illegible] $V_n = U_nP_n^{-1}B_nU_n^{-1}$ $W_n = U_nP_n^{-1}C_nU_n^{-1}$ [illegible]
2. $\{P_n^{-1}B_n\}_n$ [illegible] $r \in \mathbb{C}$ [illegible] $\rho(P_n^{-1}B_n)$ [illegible] $b$ [illegible] $b \geq 0$ [illegible] $n$.
3. $\{P_n^{-1}C_n\}_n$ [illegible] $s \in \mathbb{C}$ [illegible] $\rho(P_n^{-1}C_n)$ [illegible] $c$ [illegible] $c \geq 0$ [illegible] $n$

[illegible] $\{P_n^{-1}A_n\}_n$ [illegible] $r + s$ [illegible] $\rho(P_n^{-1}A_n)$ [illegible] $b + c$

[illegible] Since we are interested in the eigenvalues of $P_n^{-1}A_n$, it is natural to consider $U_nP_n^{-1}A_nU_n^{-1}$ which is similar to the original matrix. Moreover,

$$U_nP_n^{-1}A_nU_n^{-1} = V_n + W_n = (r+s)I_n + (V_n - rI_n) + (W_n - sI_n), \quad I_n \text{ identity matrix.}$$

By items 2 and 3 it is evident that $\{V_n - rI_n\}_n$ and $\{W_n - sI_n\}_n$ are both properly clustered at zero in the eigenvalue sense. Moreover, $V_n$ and $W_n$ are normal (item 1) and so are $V_n - rI_n$ and $W_n - sI_n$: as a consequence, $\{V_n - rI_n\}_n$ and $\{W_n - sI_n\}_n$ are also both properly clustered at zero in the singular value sense (the singular values are the moduli of the eigenvalues). Moreover, by the triangle inequality and from the assumption on the spectral radii, we have

$$\|V_n - rI_n\|_2 \leq |r| + \|V_n\|_2 = |r| + \rho(P_n^{-1}B_n) \leq |r| + b$$

and

$$\|W_n - sI_n\|_2 \leq |r| + \|W_n\|_2 = |s| + \rho(P_n^{-1}C_n) \leq |s| + c.$$

Finally, the matrix sequence

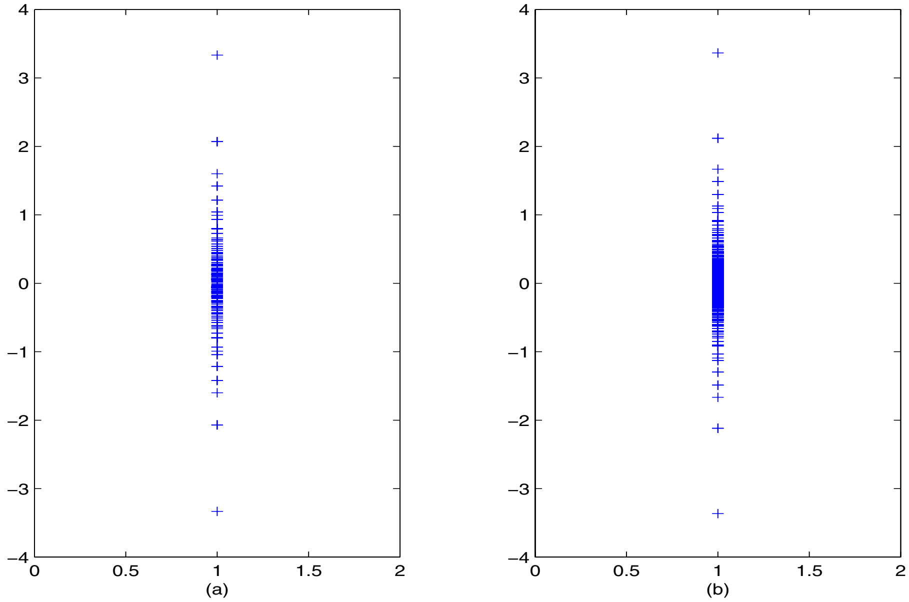
$$\{Z_n = V_n - rI_n + W_n - sI_n\}_n$$

is properly clustered at zero in the singular value sense (by the singular value decomposition) and its spectral norm is bounded, by the triangle inequality, by $|r| + b + |s| + c$ which is independent of $n$. Therefore, by Theorem 4.2, the sequence $\{Z_n\}_n$ is properly clustered at zero in the eigenvalue sense and $\{P_n^{-1}A_n\}_n$ is properly clustered at $r + s$ in the eigenvalue sense with $\rho(P_n^{-1}A_n) \leq |r + s| + |r| + b + |s| + c$. However, by exploiting again similarity and normality, the latter estimate can be substantially improved (leading to a more natural estimate) by observing that

$$\begin{aligned}\rho(P_n^{-1}A_n) = \rho(V_n + W_n) &\leq \|V_n + W_n\|_2 \leq \|V_n\|_2 + \|W_n\|_2 \\ &= \rho(P_n^{-1}B_n) + \rho(P_n^{-1}C_n) \leq b + c. \qquad \square\end{aligned}$$

It is worth mentioning that the latter result is an extension (potentially for nonsymmetric preconditioners) of Proposition 2.1 in [24]. Moreover, Theorem 4.3 works unchanged if the assumption of normality of $X_n \in \{V_n, W_n\}$ is replaced with a weaker one such as the existence of a pure constant $d \geq 1$ (independent of $n$) such that for all $j$ and uniformly with respect to $n$ it holds that $\sigma_j \leq d|\lambda_j|$, where the values $\lambda_j$ and $\sigma_j$ are the eigenvalues and the singular values of $X_n$, respectively, arranged by nondecreasing moduli.

Corollary 4.4. [illegible] 4.1 [illegible] $\{P^{-1}A\}_n$ [illegible] $1 \in \mathbb{C}^+$ $\mathbb{C}^+$ [illegible]



(a)

(b)

FIG. 4.1. *Eigenvalues for the preconditioned problem with* $\nu = 1/30$, $a = 1$, *discretization in two dimensions using centered differences and* $q = [-\sqrt{2}/2 \quad \sqrt{2}/2]^T$. (a) $h = 1/16$; (b) $h = 1/32$, *h stepsize.*

[illegible]

[illegible] The localization result simply follows from Bendixson (see, e.g., [17]): indeed, it is clear that any eigenvalue of $P^{-1}A$ has to belong to the field of values

$$\mathcal{F} = \left\{ z \in \mathbb{C} : \, z = \frac{x^*\mathrm{Re}(A)x}{x^*Px} + \mathrm{i}\frac{x^*\mathrm{Im}(A)x}{x^*Px}, \ x \in \mathbb{C}^n \backslash \{0\} \right\} \tag{4.5}$$

and that any eigenvalue of $P^{-1}\mathrm{Re}(A)$ and any eigenvalue of $P^{-1}\mathrm{Im}(A)$ must stay in

$$\left\{ z \in \mathbb{C} : \, z = \frac{x^*\mathrm{Re}(A)x}{x^*Px}, x \in \mathbb{C}^n \backslash \{0\} \right\} \text{ and } \left\{ z \in \mathbb{C} : \, z = \frac{x^*\mathrm{Im}(A)x}{x^*Px}, x \in \mathbb{C}^n \backslash \{0\} \right\},$$

respectively. Therefore, from Theorems 3.1 and 3.2 we deduce that all the eigenvalues of $P^{-1}A$ belong to $\{z \in \mathbb{C} : \ \mathrm{Re}(z) \in [c, C], \mathrm{Im}(z) \in [-\hat{c}, \hat{c}]\}$ with $c, C > 0$, $\hat{c} \geq 0$ independent of the dimension $n$, as in Theorems 3.1 and 3.2.

Now setting $U_n = P^{1/2}$, $P_n = P$, and $A_n = A$ we have (a) the eigenvalues of $\{P^{-1}\mathrm{Re}(A)\}_n$ are properly clustered to 1 and all lie in a uniformly bounded interval (Theorem 3.1), and $V_n = P^{-1/2}\mathrm{Re}(A)P^{-1/2}$ is symmetric and therefore normal; (b) the eigenvalues of $\{P^{-1}\mathrm{Im}(A)\}_n$ are properly clustered to 0 and all lie in a uniformly bounded interval (Theorem 3.2), and $W_n = \mathrm{i}P^{-1/2}\mathrm{Im}(A)P^{-1/2}$ is skew-symmetric and therefore normal.

Statements (a) and (b) are the assumptions of Theorem 4.3 from which we deduce that the eigenvalues of $\{P^{-1}A\}_n$ are properly clustered at $1 \in \mathbb{C}^+$. $\square$

Figures 4.1 and 4.2 report some examples of the spectrum of the coefficient matrix associated with equation (1.1) in two dimensions after preconditioning. Note the presence of the cluster in 1 in the complex field.

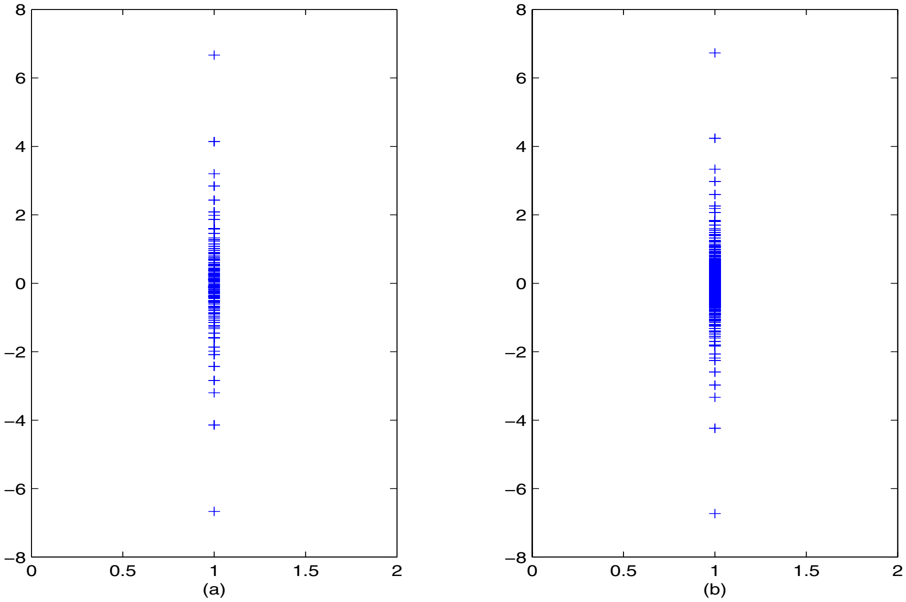


FIG. 4.2. *Eigenvalues for the preconditioned matrix with* $\nu = 1/60$*,* $a = 1$*, discretization in two dimensions using centered differences and* $q = [-\sqrt{2}/2 \quad \sqrt{2}/2]^T$. (a) $h = 1/16$*;* (b) $h = 1/32$*,* $h$ *stepsize.*

## 5. Spectrum of the preconditioned matrix.

We state here some a-priori bounds on the spectrum of the underlying preconditioned matrix.

In what follows, the numbers $\gamma_j$, $j \in \mathbb{N}$, denote constants of the order of unity, and $\alpha_d$ is defined as in section 2 (see Theorem 2.3). All these constants, in general, depend on the discretization and on the dimension $d$ of the considered domain $\Omega$. To simplify the notation, here we will focus on the two-dimensional case, where $\Omega$ is the rectangle $[0,1] \times [0,1]$. The extension to any connected finite union of rectangles in any $d$ dimension and therefore for the three-dimensional case (by just changing some constants) can be performed with the same arguments. In the result below, $d = 2$ and centered differences are used for (1.1), $\gamma_1 \to 2$ for $n \to \infty$, and $\gamma_j \to 1$ $j = 2, 3$. As usual, with $\lambda_j(X)$ we denote the generic eigenvalue of a square matrix $X$.

THEOREM 5.1. [illegible] 4.1 $\lambda_j\left(P^{-1}\mathrm{Re}(A)\right)$ [illegible]

[illegible]

$$\left[\frac{\min_{x\in\overline{\Omega}}(a)}{\max_{x\in\overline{\Omega}}(a)} - \frac{1}{\nu}\frac{\alpha_d}{2\gamma_2\pi^2}\frac{\|\nabla\cdot q\|_\infty}{\min_{x\in\overline{\Omega}}(a)}, \frac{\max_{x\in\overline{\Omega}}(a)}{\min_{x\in\overline{\Omega}}(a)} + \frac{1}{\nu}\frac{\alpha_d}{2\gamma_2\pi^2}\frac{\|\nabla\cdot q\|_\infty}{\min_{x\in\overline{\Omega}}(a)}\right]. \tag{5.1}$$

[illegible]

$$\left|\lambda_j\left(P^{-1}\mathrm{Im}(A)\right)\right| \in \left[0, \left(1+\pi^{-3}\right)\frac{\alpha_d}{\nu}\gamma_1\|q\|_\infty\frac{\max_{x\in\overline{\Omega}}(a)}{[\min_{x\in\overline{\Omega}}(a)]^2}\right]. \tag{5.2}$$

[illegible] By (4.5) and the properties of the field of values, we have

$$\mathrm{Re}\left(\lambda_j\left(P^{-1}A\right)\right) \in \left[\min_{x\in\mathbb{C}^n\setminus\{0\}}\frac{x^*\mathrm{Re}(A)x}{x^*Px}, \max_{x\in\mathbb{C}^n\setminus\{0\}}\frac{x^*\mathrm{Re}(A)x}{x^*Px}\right] \tag{5.3}$$

and

$$\operatorname{Im}\left(\lambda_j\left(P^{-1}A\right)\right) \in \left[\min_{x\in\mathbb{C}^n\setminus\{0\}} \frac{x^*\operatorname{Im}(A)x}{x^*Px}, \max_{x\in\mathbb{C}^n\setminus\{0\}} \frac{x^*\operatorname{Im}(A)x}{x^*Px}\right]. \tag{5.4}$$

For the sake of clarity, we prove the statements through three progressive steps.

- Let $a \in \mathbb{R}$ and $q \in \mathbb{R}^d$ be constants in (1.1). Then, $P \equiv \operatorname{Re}(A)$ and

  $$P^{-1}A = I + \mathrm{i}P^{-1}\operatorname{Im}(A).$$

  Therefore, the real part of the eigenvalues of the preconditioned matrix is equal to 1. Moreover, by using similar arguments as in [5, Theorem 3.2], we have the following bound for $\lambda_j(P^{-1}\operatorname{Im}(A))$:

  $$\left|\lambda_j\left(P^{-1}\operatorname{Im}(A)\right)\right| \in \left[0,\ \frac{1}{\nu}\|q\|_\infty \cdot \left(1+\pi^{-3}\right)\gamma_1\right].$$

- Let $q(x) = q$ be constant and $a(x) > 0$ in (1.1). The discretization of the diffusive part $\Theta(a)$ is exactly $\operatorname{Re}(A)$. Therefore, by [23, Theorem 8.1],

  $$\lambda_j\left(P^{-1}\operatorname{Re}(A)\right) \in \left[\frac{\min_{x\in\overline{\Omega}}(a)}{\max_{x\in\overline{\Omega}}(a)},\ \frac{\max_{x\in\overline{\Omega}}(a)}{\min_{x\in\overline{\Omega}}(a)}\right]. \tag{5.5}$$

  Moreover, $\Psi(q) \equiv \mathrm{i}\operatorname{Im}(A)$ (i.e., the discretization of the convective part is exactly $\mathrm{i}\operatorname{Im}(A)$). As a consequence, by [5, Theorem 3.2, 3.3, and 3.4], we have

  $$\left|\lambda_j\left(P^{-1}\operatorname{Im}(A)\right)\right| \in \left[0, \frac{1}{\nu}\|q\|_\infty \cdot \frac{\max_{x\in\overline{\Omega}}(a)}{[\min_{x\in\overline{\Omega}}(a)]^2}\left(1+\pi^{-3}\right)\gamma_1\right]. \tag{5.6}$$

- Finally, let us consider the general case, i.e., $a(x) : \Omega \to \mathbb{R}^+$ and $q(x) : \Omega \to \mathbb{R}^d$. Recalling Theorem 2.3, we deduce

  $$\operatorname{Re}(A(a,q)) = \Theta(a) + E, \quad \mathrm{i}\operatorname{Im}(A(a,q)) = \Psi(q) - E,$$

  $$\frac{x^*\operatorname{Re}(A)x}{x^*P\,x} = \frac{x^*\Theta(a)x}{x^*P\,x} + \frac{x^*Ex}{x^*P\,x}, \tag{5.7}$$

  $$\frac{x^*\operatorname{Im}(A)x}{x^*P\,x} = \frac{x^*\Psi(q)x}{x^*P\,x} - \frac{x^*Ex}{x^*P\,x}. \tag{5.8}$$

  By (3.1), we observe that

  $$\min\lambda_j(P) \geq 2\gamma_2\pi^2h^2\min_{x\in\overline{\Omega}}(a), \quad \max\lambda_j(P) \leq 8\gamma_3\max_{x\in\overline{\Omega}}(a),$$

  and invoking Theorem 2.3 (i.e., $\|E\|_2 \leq h^2\alpha_d\|\nabla\cdot q\|_\infty$), that

  $$\left|\frac{x^*Ex}{x^*Px}\right| \leq \frac{1}{\nu}\frac{\alpha_d}{2\gamma_2\pi^2}\frac{\|\nabla\cdot q\|_\infty}{\min_{x\in\overline{\Omega}}(a)}.$$

  Therefore, from (5.5), (5.7), and Theorem 2.3, we have (5.1). On the other hand,

  $$P^{-1}\operatorname{Im}(A) = -\frac{\mathrm{i}}{2}P^{-1}\left(\Psi(q) - \Psi(q)^*\right), \tag{5.9}$$

and hence, by the same arguments as in [5, Theorem 3.4], we deduce

$$\frac{\min_{x\in\overline{\Omega}}(a)}{[\max_{x\in\overline{\Omega}}(a)]^2}Z \le P^{-1/2}\mathrm{Im}(A)P^{-1/2} \le \frac{\max_{x\in\overline{\Omega}}(a)}{[\min_{x\in\overline{\Omega}}(a)]^2}Z, \tag{5.10}$$

with

$$Z = [A(1,\Omega)]^{-1/2}\mathrm{Im}(A)[A(1,\Omega)]^{-1/2}.$$

Finally, by the similarity of the two sequences of matrices

$$\left\{P^{-1}\mathrm{Im}(A)\right\}_n \text{ and } \left\{P^{-1/2}\mathrm{Im}(A)P^{-1/2}\right\}_n,$$

and considering expressions (5.9), (5.10), and (5.8), Theorems 3.2 and 2.3, and [5, Theorem 3.2], we infer (5.2), i.e., the desired result. □

If $q(x)$ is not a constant function, we note that the eigenvalues of the spectrum of the preconditioned matrix can have negative real part if $||\nabla\cdot q||_\infty$ is huge and/or $\nu$ is small. This may slow down the initial phase of the convergence process of the Krylov subspace projection method used to solve the underlying preconditioned linear system. However, if the convection is overly dominant, a preconditioning strategy based on a suitable upwind discretization can be used. The related eigenvalue analysis can be adapted by using tools similar to those considered here.

## 6. Notes on the convergence of iterative methods.

**6.1. The condition number of the eigenvector matrix.** Here we will focus on the case

$$q = [\cos(\phi) \quad \sin(\phi)]^T, \quad 0 \le \phi < \pi,$$

where $\phi$ is a constant angle; i.e., the wind is constant. In this case, the following result holds true. For simplicity, here we focus on the case when $N_1 = N_2 = \cdots = N_d = n^{1/d}$, where $n$ is the size of $A$ (uniform grid).

LEMMA 6.1. [illegible] $q(x)$ [illegible] $a(x)$ [illegible] (1.1) [illegible] (1.1) [illegible] $P^{-1}A$ [illegible] $n$ [illegible] $V$ [illegible] $P^{-1}A$ $V$ [illegible] $\kappa_2(V) \sim n^{1/d}$ [illegible] $N_i \approx \alpha_{i,j}N_j$ [illegible] $i, j$, [illegible] $\alpha_{i,j}$ [illegible] $\kappa_2(V) \approx cn^{1/d}$ [illegible] $c$ [illegible]

[illegible] Under our assumptions, since $q(x)$ and $a(x)$ in (1.1) are constant, then $P \equiv \Theta(1)$ and $\Psi(q)$ is a skew-symmetric matrix. Moreover, the preconditioned matrix $P^{-1}A$ can be written as

$$P^{-1}A = (\Theta(1))^{-1}\cdot(\Theta(1) + \Psi(q)) = I + (\Theta(1))^{-1}\Psi(q) = I + (\Theta(1))^{-1/2}S(\Theta(1))^{1/2},$$

where $\Theta(1)$ and $\Psi(q)$ are the matrices generated by the discretization of the diffusive and convective parts of (1.1), respectively. However, by construction $S = (\Theta(1))^{-1/2}\,\Psi(q)\,(\Theta(1))^{-1/2}$ is a skew-symmetric matrix since $(\Theta(1))^{-1/2}$ is a symmetric positive definite matrix and $\Psi(q)$ is a skew-symmetric matrix. Therefore, $I+S$ is normal because

$$(I+S)^*\cdot(I+S) = (I-S)(I+S) = I - S^2,$$

which is the same matrix obtained as $(I+S)\cdot(I+S)^*$. Consequently,

$$\begin{aligned} P^{-1}A &= (\Theta(1))^{-1/2}(I+S)(\Theta(1))^{1/2} \\ &= (\Theta(1))^{-1/2}QDQ^*(\Theta(1))^{1/2}, \end{aligned}$$

where $D$ is diagonal (the eigenvalue matrix), $Q$ is unitary, and $V=(\Theta(1))^{-1/2}Q$ is the eigenvector matrix. Since $\kappa_2(P^{-1})=\kappa_2((\Theta(1))^{-1})\sim n^{2/d}$ (it is a classical result on the discrete Laplacian; refer, e.g., to part 3 of Theorem 2.1), it directly follows that $\kappa_2(V)=\kappa_2((\Theta(1))^{-1/2}Q)=\kappa_2((\Theta(1))^{-1/2})\sim n^{1/d}$. Moreover, if $N_i\approx\alpha_{i,j}N_j$ for every $i,j$ and $\alpha_{i,j}$ are universal constants, then $\kappa_2((\Theta(1))^{-1})\approx c^2n^{2/d}$, where $c$ is a pure positive constant, and therefore $\kappa_2(V)=\kappa_2((\Theta(1))^{-1/2}Q)=\kappa_2((\Theta(1))^{-1/2})\approx cn^{1/d}$. □

Note that if we could use $P^{-1/2}$ as a split preconditioner instead of $P$ as a left (or right) preconditioner, then $\kappa_2(V)=1$ because $P^{-1/2}AP^{-1/2}=I+S$ is normal. This, in theory, could have some relevance for the convergence (see the next section); in practice we observed no changes.

**6.2. Analysis of the convergence.** To study the convergence of GMRES, we report a few tools based on polynomials related to the minimal polynomial of the matrix $K$ of the underlying linear system, which have been introduced in [10].

Recall the bound on the convergence of GMRES (see [20, sections 6.11.2, 6.11.4]):

$$||r_j||_2 \le \kappa_2(V)\cdot \min_{p_j(0)=1}\max_{\lambda\in\lambda(K)}|p_j(\lambda)|\cdot\|r_0\|_2, \tag{6.1}$$

where $\lambda(K)$ is the set of all the eigenvalues of the matrix $K$, $\kappa_2(V)$ is the spectral condition number of the matrix of the eigenvectors of $K$, $V$ is chosen to minimize $\kappa_2(V)$, is and $p_j(z)$ is a polynomial of degree at most $j$. Note that, under the assumptions of Lemma 6.1, we have $\kappa_2(V)=cn^{1/d}$, with $c$ a universal constant.

Let us consider the preconditioned sequence $\{K=P^{-1}A\}_n$ whose spectrum $\{\lambda(K)\}_n$ is clustered (recall Corollary 4.4) and partition $\lambda(K)$ as in [4]:

$$\lambda(K)=\lambda^{(c)}(K)\cup\lambda^{(0)}(K)\cup\lambda^{(1)}(K),$$

where $\lambda^{(c)}(K)$ denotes the clustered set of eigenvalues of $K$ and $\lambda^{(0)}(K)\cup\lambda^{(1)}(K)$ denotes the set of the (distinct) outliers. We assume that the clustered set $\lambda^{(c)}(K)$ of eigenvalues is contained in a convex set $\mathcal{C}$ whose closure must not contain the origin.

The sets

$$\lambda^{(0)}(K)=\{\hat\lambda_1,\hat\lambda_2,\ldots,\hat\lambda_{j_0}\}\quad\text{and}\quad\lambda^{(1)}(K)=\{\tilde\lambda_1,\tilde\lambda_2,\ldots,\tilde\lambda_{j_1}\}$$

denoting two sets of $j_0$ and $j_1$ outliers, respectively, are defined as in [4]; i.e., if $\hat\lambda_j\in\lambda^{(0)}(K)$, we have

$$1<\left|1-\frac{z}{\hat\lambda_j}\right|\le c_j\quad\forall z\in\mathcal{C},$$

while, for $\tilde\lambda_j\in\lambda^{(1)}(K)$,

$$0<\left|1-\frac{z}{\tilde\lambda_j}\right|<1\quad\forall z\in\mathcal{C},$$

respectively.

From (6.1) and the above definitions, we have

$$\min_{p_j(0)=1} \max_{z\in\lambda(K)} |p_j(z)| \leq \max_{z\in\lambda(K)} |\hat{p}(z)\cdot q(z)\cdot \tilde{p}(z)|, \tag{6.2}$$

where

$$\hat{p}(z) = \left(1-\frac{z}{\hat{\lambda}_1}\right)\cdots\left(1-\frac{z}{\hat{\lambda}_{j_0}}\right), \quad \tilde{p}(z) = \left(1-\frac{z}{\tilde{\lambda}_1}\right)\cdots\left(1-\frac{z}{\tilde{\lambda}_{j_1}}\right)$$

are the polynomials whose roots are the (distinct) outlying eigenvalues in $\lambda^{(0)}(K) \cup \lambda^{(1)}(K)$ and $q(z)$ is a polynomial of degree at most $j - j_0 - j_1 \geq 0$ such that $q(0) = 1$. The polynomial $q(z)$ can be chosen to be the shifted and scaled complex Chebyshev polynomial $q(z) = C_k((c-z)/d)/C_k(c/d)$ which is small on the set containing $\lambda^{(c)}(K)$; see [20, sections 6.11.2, 6.11.4]. Therefore, by using the same arguments as in [4], we have the following.

THEOREM 6.2. [illegible] $j$ [illegible] $\epsilon$ [illegible] $2$ [illegible] $||r_j||_2/||r_0||_2$ [illegible] $Kx = b$ $K$ [illegible]

$$\min\left\{ j_0 + j_1 + \left\lceil \frac{\log(\epsilon) - \log(\kappa_2(V))}{\log(\rho)} - \sum_{\ell=1}^{j_0} \frac{\log(c_\ell)}{\log(\rho)} \right\rceil, n \right\}, \tag{6.3}$$

[illegible]

$$\rho^k = \frac{\left(a/d + \sqrt{(a/d)^2-1}\right)^k + \left(a/d + \sqrt{(a/d)^2-1}\right)^{-k}}{\left(c/d + \sqrt{(c/d)^2-1}\right)^k + \left(c/d + \sqrt{(c/d)^2-1}\right)^{-k}}, \tag{6.4}$$

[illegible] $\mathcal{C} \in \mathbb{C}^+$ [illegible] $c$ [illegible] $d$ [illegible] $a$

The bound (6.3) suggests that there will be a latency of $j_0 + j_1$ steps before the asymptotic behavior is observed. If $j_0 > 0$, then there may be some additional delay proportional to $(\sum_l \log c_l)^{-1}$. In practice, the asymptotic convergence behavior will not be manifested until the expression

$$\max_{z\in\lambda^{(c)}(K)} |\hat{p}(z)\cdot\tilde{p}(z)|\rho^k$$

is less than 1, where $k$ is the degree of the shifted and scaled Chebyshev polynomial. Of course, these are theoretical arguments because $||p_j||$ can be arbitrarily large, and then no general statements can be made about how much larger the delay in convergence can be in practice or when superlinear convergence sets in.

### 6.3. Examples and comments.

In this section we report on a few experiments with a centered difference discretization and constant coefficients for problem (1.1) in order to compare the theoretical results and notes above. The preconditioner $P$ is implemented here in MATLAB by using a fast Poisson solver. Performances (timings) can be improved with a multigrid-based fast Poisson solver, but this will be considered in a future work together with more general test problems. Experiments are performed with GMRES but we include also two-dimensional tests and (total) timings with preconditioned and nonpreconditioned BiCGSTAB. In three or more dimensions,

TABLE 6.1

*Preconditioned GMRES iterations for centered differences discretization of* (1.1)*, two-dimensional problem,* $q = [-\sqrt{2}/2 \quad \sqrt{2}/2]^T$*,* $a = 1$*,* $\epsilon = 10^{-6}$*. In parentheses: nonpreconditioned (full) GMRES iterations.*

| $h \setminus \nu$ | 1/10 | 1/20 | 1/30 | 1/40 | 1/60 | 1/80 |
|---|---|---|---|---|---|---|
| 1/16 | 11(31) | 18 (29) | 23 (29) | 27 (31) | 35 (31) | 44 (31) |
| 1/32 | 11 (47) | 17 (51) | 23 (51) | 27 (54) | 36 (58) | 47 (61) |
| 1/64 | 10 (52) | 15 (57) | 21 (75) | 25 (85) | 35 (97) | 45 (106) |
| 1/128 | 8 (51) | 13 (52) | 19 (54) | 23 (55) | 31 (77) | 43 (109) |

TABLE 6.2

*Preconditioned matrix-vector products* ($2 \times$ *iterations*) *for BiCGSTAB on centered differences discretization of* (1.1)*, two-dimensional problem,* $q = [-\sqrt{2}/2 \quad \sqrt{2}/2]^T$*,* $a = 1$*,* $\epsilon = 10^{-6}$*. In parentheses: nonpreconditioned BiCGSTAB matrix-vector products.*

| $h \setminus \nu$ | 1/10 | 1/20 | 1/40 | 1/60 | 1/80 |
|---|---|---|---|---|---|
| 1/128 | 11 (447) | 17 (427) | 39 (483) | 61 (499) | 93 (400) |
| 1/256 | 9 (786) | 17 (817) | 36 (929) | 56 (967) | 80 (981) |
| 1/512 | 6 (785) | 15 (1609) | 31 (1935) | 47 (1953) | 71 (1963) |
| 1/1024 | 5 (1873) | 13 (†) | 25 (†) | 42 (†) | 59 (†) |

TABLE 6.3

*Timings (in seconds) for BiCGSTAB on centered differences discretization of* (1.1)*, two-dimensional problem,* $q = [-\sqrt{2}/2 \quad \sqrt{2}/2]^T$*,* $a = 1$*,* $\epsilon = 10^{-6}$*. In parentheses: nonpreconditioned BiCGSTAB timings. Note that halving the stepsize means that the sizes of matrices are multiplied by four.*

| $h \setminus \nu$ | 1/10 | 1/20 | 1/40 | 1/60 | 1/80 |
|---|---|---|---|---|---|
| 1/128 | 1.2 (1.5) | 4.5 (1.5) | 4.3 (1.3) | 6.2 (1.6) | 8.8 (1.2) |
| 1/256 | 3. (13.1) | 6.2 (13.53) | 31.1 (15.9) | 20 (16) | 30.9 (17) |
| 1/512 | 7.9 (111) | 19.7 (113) | 40.4 (138) | 56 (145) | 82.3 (139) |
| 1/1024 | 27.28 (1019) | 62.2 (†) | 120.5 (†) | 191 (†) | 248 (†) |

fair timings require a more efficient implementation. For memory limitations, we provide large tests for BiCGSTAB only. A dagger † in the tables means that the solver does not converge after 1000 iterations (i.e., 1000 matrix-vector products for GMRES and 2000 for BiCGSTAB).

Our experiments are performed under the assumptions of Lemma 6.1. By Theorem 5.1, we have $j_0 = 0$. Therefore, the delay for asymptotic convergence behavior is mainly related to the number of distinct outlying eigenvalues. However, if $\epsilon$ is large enough, GMRES may treat as multiple eigenvalues those which belong to $\lambda^{(1)}$, are nondefective, and form small satellite clusters, as observed in [10]. In this case, the above mentioned delay can be less than $j_1$ iterations.

We stress that the presence of a proper cluster of eigenvalues means also that the number of the outliers does not increase with $N$, provided that it is large enough, and that their influence is limited to an initial delay for the asymptotic phase of convergence.

In Tables 6.1, 6.2, and 6.3, we report the number of preconditioned and nonpreconditioned GMRES iterations for the underlying two-dimensional problem with

$$q = [-\sqrt{(}2)/2 \ \sqrt{(}2)/2]^T,$$

$a = 1$, $\epsilon = 10^{-6}$ for $h = 1/16$ to $h = 1/128$, and $\nu = 1/10$ to $\nu = 1/80$, and similarly for BiCGSTAB. The boundary conditions in (1.1) are

$$u(0, y) = u(1, y) = 1, \quad 0 < y < 1; \quad u(x, 0) = u(x, 1) = 0, \quad 0 < x < 1.$$

Table 6.4

*Preconditioned GMRES iterations for centered differences discretization of* (1.1)*, three-dimensional problem with* $q = [1/\sqrt{3} \quad 1/\sqrt{3} \quad 1/\sqrt{3}]^T$*,* $a = 1$*,* $\epsilon = 10^{-6}$*. In parentheses: non-preconditioned (full) GMRES iterations.*

| $h \setminus \nu$ | 1/10 | 1/20 | 1/30 | 1/40 | 1/60 | 1/80 |
|---|---|---|---|---|---|---|
| 1/8 | 12 (25) | 17 (23) | 22 (21) | 26 (25) | 33 (31) | 40 (37) |
| 1/16 | 12 (51) | 18 (50) | 24 (48) | 29 (47) | 38 (45) | 49 (50) |
| 1/32 | 11 (93) | 17 (97) | 23 (97) | 28 (97) | 38 (95) | 49 (94) |

In Table 6.4 we report similar tests with the three-dimensional problem using GMRES but with

$$q = [1/\sqrt{3} \quad 1/\sqrt{3} \quad 1/\sqrt{3}]^T,$$

and the boundary conditions are $u(0,0,0) = 1$ and zero elsewhere. Similar results are obtained with other Dirichlet boundary conditions.

We note that halving the stepsize means that the sizes of matrices are multiplied by four. The theoretical computational cost is $O(N)$, where the mesh is equispaced, and thus $N = n^d$, with $d$ the dimension of the domain. However, we can see that when we halve the stepzise, timings for preconditioned iterations (see Table 6.3, where $d = 2$) are always less than quadruple.

**6.4. Convergence and the viscosity parameter.** In the analysis performed in section 5 we observed that, if $q$ in (1.1) is constant, then the imaginary parts of the eigenvalues of the preconditioned matrix are proportional to $\nu^{-1}$; see Theorem 5.1. Moreover, the number of the distinct outliers does not depend on $\nu$ or on the mesh, but it does depend on the choice of the function $q$; see the results on the existence of a proper cluster in the previous sections. For example, if $a(x)$ is also constant, we have

$$\beta = \max_j \{ \left| \operatorname{Im} \left( \lambda_j (P^{-1}A) \right) \right| \} = \frac{c}{\nu},$$

where $c$ is a universal positive constant. Another evidence of this can be found in Figures 4.1 and 4.2.

Moreover, by denoting with $\beta$ the radius of the cluster and provided that $\beta > 0$, with the notation of Theorem 6.2, the contribution to the number of the iterations of the eigenvalues in the cluster is bounded from above by

$$\frac{\log(\epsilon)}{\log(\rho)} = c' \frac{\log(\epsilon)}{\frac{-1}{1+\beta}} = c'(1+\beta)\log(\epsilon^{-1}). \tag{6.5}$$

Here, $c'$ is a pure positive constant which takes into account that $\rho$ is approximated by

$$\tilde{\rho} = \frac{\beta}{1 + \sqrt{1+\beta^2}} < \frac{\beta}{1+\beta} = 1 - \frac{1}{1+\beta}$$

and that, provided that $\beta > 0$, $\log(\rho)$ is approximated by the Taylor expansion of $\log(\tilde{\rho})$, with $\tilde{\rho}$ being defined as above. Again, note that we are in the hypotheses of Lemma 6.1, and then the convergence is dictated by the distribution of the eigenvalues. Therefore, the number of iterations is expected to grow with $\nu^{-1}$. However, in practice, the number of iterations seems to be proportional to $\sqrt{\nu^{-1}}$ (see Tables 6.1

and 6.4), and this behavior is confirmed for various functions $a(x)$ and $q(x)$; see also the numerical experiments in [6].

The above discussion was done under restrictive hypotheses. However, the experience of several different choices of functions $a(x)$ and $q(x)$ and values of the viscosity parameter $\nu$ (always such that the hypotheses of Theorem 4.1 are satisfied) suggests that the number of the iterations depends on a function of $\nu^{-1}$, even under more general assumptions, but it is independent of the mesh and of the dimension $d$ of problem (1.1).

## 7. Conclusions.

The purpose of this work was to explore some properties of the preconditioned operator $P^{-1}A$, where $P$ is defined in (3.1) and $A$ is the matrix generated by a finite difference discretization (using centered differences or upwind) of the convection-diffusion equation (1.1). In particular, we proved the existence of a cluster in the spectrum of $\{P^{-1}A\}_n$ and gave a bound for the condition number of the matrix of the eigenvector. Moreover, we found that eigenvalue distribution and convergence rates are independent of the discretization mesh size and of the dimension of the problem but do depend (weakly) on $\nu^{-1}$.

Indeed, beside the spectral theoretical analysis of the preconditioned structures, we stress that our technique can be easily implemented. In fact, the ingredients are constituted by the following blocks: a Krylov method (e.g., GMRES, BiCGSTAB, etc.), a matrix vector routine (for sparse or even diagonal matrices), and a solver for the related diffusion equation with a constant coefficient (a method based, e.g., on the cyclic reduction approach [9, 14] or on multigrid methods [27, 19] for which professional software is available). Of course, if the convection part is dominating, then the considered approach can be enriched by alternating the discussed diffusion-based preconditioning with a preconditioner for an upwind discretization. At this point, we recall that the idea of using, e.g., a multigrid (for a simpler differential problem) as a preconditioner in a Krylov-type method is quite classical, as it emerges in [18, 27]. In this direction, we must quote the following statements from Greenbaum [18, subsection 12.1.5, p. 197]:

> Some multigrid aficionados will argue that if one has used the proper restriction, prolongation, and relaxation operators, then the multigrid algorithm will require so few cycles ... that it is almost pointless to try to accelerate it with CG-like methods. This may be true, but unfortunately such restriction, prolongation, and relaxation schemes are not always known. In such cases, CG, GMRES QMR, or BiCGSTAB acceleration may help.
>
> Equivalently, one can consider multigrid as a preconditioner for one of these Krylov subspace methods.

A future work will be in the direction of combining different iterative solvers (the multi-iterative idea [21]) and more specifically we would like (A) to use the preconditioner considered in this paper as one of the smoothers for a V-cycle directly in the original problem; (B) to make a comparison between the present approach and the one in (A); and (C) to enrich the analysis in the case of convection-dominated problems in order to achieve more robustness.

**Acknowledgments.** We are grateful to the referees for useful comments which have improved this presentation.

## REFERENCES

[1] Z. Z. Bai, G. H. Golub, and M. K. Ng, *Hermitian and Skew-Hermitian splitting methods for non-Hermitian positive definite linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 603–626.

[2] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 20–41.

[3] D. Bertaccini and M. K. Ng, *The convergence rate of block preconditioned systems arising from LMF-based ODE codes*, BIT, 41 (2001), pp. 433–450.

[4] D. Bertaccini and Michael K. Ng, *Band-Toeplitz preconditioned GMRES iterations for time-dependent PDEs*, BIT, 40 (2003), pp. 901–914.

[5] D. Bertaccini, G. H. Golub, S. Serra-Capizzano, and C. Tablino-Possio, *Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation*, Numer. Math., 99 (2005), pp. 441–484.

[6] D. Bertaccini, G. H. Golub, and S. Serra-Capizzano, *Analysis of a Preconditioned Iterative Method for the Convection-Diffusion Equation*, preprint SCCM-03-13, Stanford University, Stanford, CA, 2003. Available online at http://www-sccm.stanford.edu/wrap/pub-tech.html.

[7] R. Bhatia, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[8] A. Böttcher and B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices*, Springer-Verlag, New York, 1998.

[9] B. L. Buzbee, G. H. Golub, and C. W. Nielson, *On direct methods for solving Poisson's equations*, SIAM J. Numer. Anal., 7 (1970), pp. 627–656.

[10] S. L. Campbell, I. C. F. Ipsen, C. T. Kelley, and C. D. Meyer, *GMRES and the minimal polynomial*, BIT, 36 (1996), pp. 664–675.

[11] P. Concus and G. H. Golub, *Use of fast direct methods for the efficient numerical solution of nonseparable elliptic equations*, SIAM J. Numer. Anal., 10 (1973), pp. 1103–1120.

[12] P. Concus and G. Golub, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, in Computing Methods in Applied Sciences and Engineering, Lecture Notes in Econom. and Math. Systems 134, Springer, Berlin, 1976, pp. 56–65.

[13] P. Concus, G. H. Golub, and G. Meurant, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 220-252.

[14] F. W. Dorr, *The direct solution of the discrete Poisson equation on a rectangle*, SIAM Rev., 12 (1970), pp. 248–263.

[15] H. C. Elman and M. H. Schultz, *Preconditioning by fast direct methods for nonself-adjoint nonseparable elliptic equations*, SIAM J. Numer. Anal., 23 (1986), pp. 44–57.

[16] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations*, Numer. Math., 90 (2002), pp. 641–664.

[17] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[18] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.

[19] W. Hackbusch, *Multigrid Methods and Applications*. Springer-Verlag, Berlin, 1985.

[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[21] S. Serra Capizzano, *Multi-iterative methods*, Comput. Math. Appl., 26 (1993), pp. 65–87.

[22] S. Serra Capizzano, *On the extreme eigenvalues of Hermitian (block) Toeplitz matrices*, Linear Algebra Appl., 270 (1998), pp. 109–129.

[23] S. Serra Capizzano, *The rate of convergence of Toeplitz based PCG methods for second order nonlinear boundary value problems*, Numer. Math., 81 (1999), pp. 461–495.

[24] S. Serra-Capizzano, D. Bertaccini, and G. H. Golub, *How to deduce a proper eigenvalue cluster from a proper singular value cluster in the nonnormal case*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 82–86.

[25] S. Serra Capizzano and C. Tablino Possio, *Preconditioning strategies for 2D finite difference matrix sequences*, Electr. Trans. Numer. Anal., 16 (2003), pp. 1–29.

[26] S. Serra Capizzano and C. Tablino Possio, *Superlinear preconditioners for finite differences linear systems*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 152–164.

[27] U. Trottenberg, C.W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[28] H. A. van der Vorst and C. Vuik, *The superlinear convergence behavior of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 279–301

# INVERSE SPECTRAL PROBLEMS FOR SEMISIMPLE DAMPED VIBRATING SYSTEMS*

PETER LANCASTER†

*This paper is dedicated to the memory of Miron Tismenetsky, a good friend and colleague*

**Abstract.** Computational schemes are investigated for the solution of inverse spectral problems for $n \times n$ real systems of the form $L(\lambda) = M\lambda^2 + D\lambda + K$. Thus, admissible sets of data concerning systems of eigenvalues and eigenvectors are examined, and procedures for generating associated (isospectral) families of systems are developed. The analysis includes symmetric systems, systems with mixed real/nonreal spectrum, systems with positive definite coefficients, and hyperbolic systems (with real spectrum). A one-to-one correspondence between Jordan pairs and structure preserving similarities is clarified. An examination of complex symmetric matrices is included.

**Key words.** inverse problems, vibrating systems, structure preserving transformations

**AMS subject classifications.** Primary, 74A15; Secondary, 15A29

**DOI.** 10.1137/050640187

## 1. Introduction.

Inverse eigenvalue problems are addressed here in the context of vibrating systems which, for our present purposes, are defined as follows.

DEFINITION 1. (vibrating) system [illegible] $n \times n$ real [illegible] $\{M, D, K\}$ [illegible] $M$ [illegible]

Many problems of physical interest also require that some or all of the coefficients $M$, $D$, $K$ be symmetric and positive definite or semidefinite (see [2] and [16], for example).

In this paper, an idea introduced in [10] is extended from the restriction to systems with purely nonreal spectrum to the full range of real and nonreal spectrum, but with the continuing limitation (seen as nonrestrictive by many in the field) to semisimple eigenvalues; i.e., an eigenvalue of multiplicity $m \geq 1$ has $m$ associated linearly independent eigenvectors. This hypothesis has the added advantage that analysis is simplified considerably. The generation of real systems, real symmetric systems, and systems with positive semidefinite or positive definite coefficients will be considered, in this order.

First, it is necessary to summarize the spectral properties admitted in this analysis. Since the systems of interest have real coefficients, the eigenvalues may be real or may appear in complex conjugate pairs. All eigenvalues are required to be semisimple for both real and nonreal eigenvalues. The set of all the eigenvalues, both real and complex, is denoted by $\sigma$.

The central problem considered here is, given an admissible set of spectral data (real and complex eigenvalues and, where required, a sign characteristic), to construct a family of systems consistent with this data. Another closely related problem is,

---

*Received by the editors September 12, 2005; accepted for publication (in revised form) by Q. Ye August 4, 2006; published electronically February 9, 2007. Much of this work was done while the author held an Honorary Research Fellowship in the School of Mathematics at the University of Manchester. The author gratefully acknowledges partial support from the Natural Sciences and Engineering Research Council of Canada.

http://www.siam.org/journals/simax/29-1/64018.html

†Department of Mathematics and Statistics, University of Calgary, Calgary, AB, T2N 1N4, Canada (lancaste@ucalgary.ca).

given a real system (with spectral data implicitly defined), to construct a family of systems consistent with [illegible] data. In both cases the objective is the construction of [illegible] families of systems; i.e., each member of the family has spectrum $\sigma$ and it is semisimple.

For the second formulation, in particular, it is natural to reformulate the problem in terms of the well-known [illegible], in which case all eigenvalues, at least, are preserved by similarity transformations. This, in turn, leads to the notion of [illegible], which have been discussed elsewhere (in [10] and [15], for example) and which are developed further in sections 2 and 3. In section 4 these ideas are re-examined in terms of Jordan pairs (see Theorem 3), and this leads to constructions for families of real systems in section 5.

The study of [illegible] systems (in which $M$, $D$, $K$ are real and symmetric) is taken up in sections 6 and 7. A strategy is adopted in which the real eigenvectors are assigned (subject to some necessary constraints) and then the eigenvectors for [illegible] eigenvalues are determined from them. This requires the symmetric factorization of a complex symmetric matrix and is accomplished with the aid of [illegible] (section 7 and Appendix A). This also requires some detailed knowledge of the rank of complex symmetric matrices which is presented in Appendix C.

Hypotheses that ensure the positivity conditions of $M$, $D$, $K$ are the subject of section 8, where Theorem 5 is the central (new) result. Systems with all eigenvalues real (quasi hyperbolic or overdamped, for instance) are the subject of section 9, where Corollary 6 is the main contribution.

**2. Massaging the spectrum.** If the system is $n \times n$, then $2r$ real eigenvalues are admitted ($0 \le r \le n$). The nonreal eigenvalues in the upper half of the complex plane are determined by a complex diagonal matrix $\Lambda = U_1 + iW$ of size $(n-r)\times(n-r)$ with $W > 0$. Their complex conjugates are also eigenvalues and make up the diagonal entries of $\bar{\Lambda}$. Then there are $2r$ real eigenvalues which are distributed between the diagonal entries of two $r \times r$ real diagonal matrices $U_2$ and $U_3$. The way in which these two matrices are formed will be discussed in what follows.

A complex (canonical) diagonal $2n \times 2n$ matrix including all the eigenvalues is now

$$J = \begin{bmatrix} \Lambda & 0 & 0 & 0 \\ 0 & U_2 & 0 & 0 \\ 0 & 0 & U_3 & 0 \\ 0 & 0 & 0 & \bar{\Lambda} \end{bmatrix} = \begin{bmatrix} U_1 + iW & 0 & 0 & 0 \\ 0 & U_2 & 0 & 0 \\ 0 & 0 & U_3 & 0 \\ 0 & 0 & 0 & U_1 - iW \end{bmatrix}. \tag{1}$$

Defining $\Omega_1^2 = U_1^2 + W^2$, it is easily seen that there is an associated (diagonal, real symmetric) vibrating system:

$$L_0(\lambda) := \lambda^2 I_n - 2\lambda \begin{bmatrix} U_1 & 0 \\ 0 & \frac{1}{2}(U_2 + U_3) \end{bmatrix} + \begin{bmatrix} \Omega_1^2 & 0 \\ 0 & U_2 U_3 \end{bmatrix}. \tag{2}$$

It is simply a direct sum of the two diagonal systems

$$\lambda^2 I_{n-r} - 2\lambda U_1 + \Omega_1^2 = (\lambda I_{n-r} - \Lambda)(\lambda I_{n-r} - \overline{\Lambda})$$

and

$$\lambda^2 I_{2r} - \lambda(U_2 + U_3) + U_2 U_3 = (\lambda I_r - U_2)(\lambda I_r - U_3)$$

with nonreal and real eigenvalues, respectively.

Construct the abbreviations

$$U = \begin{bmatrix} U_1 & 0 \\ 0 & \frac{1}{2}(U_2+U_3) \end{bmatrix} \quad \text{and} \quad \Omega^2 = \begin{bmatrix} \Omega_1^2 & 0 \\ 0 & U_2U_3 \end{bmatrix} \tag{3}$$

so that (2) takes the form

$$L_0(\lambda) = \lambda^2 I - 2\lambda U + \Omega^2.$$

Now a particular linearization of $L_0(\lambda)$ is $\lambda I_{2n} - C_0$, where $C_0$ is the associated [illegible]:

$$C_0 := \begin{bmatrix} 0 & I_n \\ -\Omega^2 & 2U \end{bmatrix}. \tag{4}$$

Our objective is to generate vibrating systems whose companion matrices are similar to $C_0$, and which, consequently, are isospectral.

A first step in the analysis is to show that, under a weak assumption on the distribution of the [illegible] eigenvalues, an explicit similarity can be formulated which transforms the companion matrix $C_0$ into the diagonal matrix, $J$, of its eigenvalues. First, define a $2n \times 2n$ block matrix in terms of the blocks of $J$:

$$Z = \begin{bmatrix} \bar{\Lambda} & 0 & 0 & -I_{n-r} \\ 0 & -U_3 & I_r & 0 \\ 0 & -U_2 & -I_r & 0 \\ \Lambda & 0 & 0 & -I_{n-r} \end{bmatrix}. \tag{5}$$

LEMMA 1. [illegible]

$$\det(U_2 - U_3) \neq 0. \tag{6}$$

[illegible] $Z$ [illegible] $J$ [illegible] (1)

$$ZC_0Z^{-1} = J. \tag{7}$$

[illegible]. Elementary block operations can be applied to $Z$ to reduce it to the block triangular form

$$\begin{bmatrix} -2iW & 0 & I_{n-r} & 0 \\ 0 & U_2 - U_3 & 0 & I_r \\ 0 & 0 & -I_{n-r} & 0 \\ 0 & 0 & 0 & I_r \end{bmatrix}.$$

Since $W > 0$, it is apparent that $Z$ is nonsingular if and only if condition (6) is satisfied.

Then write $C_0$ in partitioned form consistent with that of $Z$:

$$C_0 = \begin{bmatrix} 0 & 0 & I_{n-r} & 0 \\ 0 & 0 & 0 & I_r \\ -\Omega_1^2 & 0 & 2U_1 & 0 \\ 0 & -U_2U_3 & 0 & U_2+U_3 \end{bmatrix}.$$

Now a simple calculation with block matrices shows that $ZC_0 = JZ$. Thus, when (6) holds, $Z$ is nonsingular and $ZC_0Z^{-1} = J$. □

Notice that condition (6), together with our standing hypotheses, ensures that the systems investigated here are [illegible] in the sense of Definition 8 of [15]. These conditions also appear in Theorem 7 of [14].

**3. Structure preserving similarities.** The following two-part definition reflects a definition introduced in the paper [10]. The underlying idea concerns similarity transformations of $C_0$ which preserve the companion matrix structure (and, necessarily, the spectrum, $\sigma$). For brevity, the term "SPS" (for structure preserving similarity) is introduced.

DEFINITION 2. [illegible] $V \in \mathbb{R}^{2n\times 2n}$ [illegible] $C_0$ [illegible]

$$C := VC_0V^{-1} = V \begin{bmatrix} 0 & I_n \\ -\Omega^2 & 2U \end{bmatrix} V^{-1} \tag{8}$$

[illegible] $C$ [illegible] $n \times n$ [illegible]

$$C = \begin{bmatrix} 0 & I_n \\ C_{21} & C_{22} \end{bmatrix}.$$

It is clear that all matrices $C$ of (8) determined by an SPS are isospectral with spectrum $\sigma$. Furthermore, the corresponding vibrating systems are isospectral [illegible] have real coefficients.

A simple lemma from [10] will be useful.

LEMMA 2 (see [10]). [illegible] $V \in \mathbb{R}^{2n\times 2n}$ [illegible] $n \times n$ [illegible] $V_{ij}$ [illegible] $C_0$ [illegible]

$$V_{21} = -V_{12}\Omega^2 \quad \text{[illegible]} \quad V_{22} = V_{11} + 2V_{12}U. \tag{9}$$

[illegible] With $V$ nonsingular, (8) is equivalent to

$$CV = V \begin{bmatrix} 0 & I_n \\ -\Omega^2 & 2U \end{bmatrix}.$$

Comparing blocks, it is found that $C_{11} = 0$ and $C_{12} = I_n$ if and only if (9) holds. □

[illegible] 1. A simple class of SPS is defined by matrices

$$V = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix},$$

where $A$ is nonsingular. These transformations generate a narrow class of systems which are similar to the canonical system and for which the coefficients $M$, $D$, $K$ commute.

[illegible] 2. Another class of SPS is generated by nonsingular matrices $V$ which commute with $C_0$. They could be described as "automorphisms" because they satisfy $VC_0V^{-1} = C_0$; they transform $C_0$ into itself. Our interest is in transformations for which the greatest possible freedom in the coefficients is achieved (consistent with preservation of the spectrum).

**4. Jordan pairs and SPS.** A right eigenvector (say, $x_j \neq 0$) can be associated with each diagonal entry of $J$ (each eigenvalue), and these form the columns of an associated $n \times 2n$ matrix of eigenvectors, say, $X$. More generally, if $X \in \mathbb{C}^{n\times 2n}$, the pair $(X, J)$ (with $J$ as in (1)) forms a [illegible] if $\left[\begin{smallmatrix} X \\ XJ \end{smallmatrix}\right]$ is nonsingular.[1] It is well known (see [4], [9], or Chapter 14 of [12], for example) that a Jordan pair, together with a mass matrix $M$, defines a system completely.

---

[1] This guarantees, in particular, that every column of $X$ (every eigenvector) is nonzero.

Here, with our hypotheses on the spectrum, we may define an $n \times 2n$ matrix of eigenvectors of $L(\lambda)$ in the form

$$X = \left[\begin{array}{cccc} X_c & X_{R1} & X_{R2} & \overline{X_c} \end{array}\right], \tag{10}$$

where $X_c$ is an $n \times (n-r)$ matrix of (generally) nonreal eigenvectors corresponding to the eigenvalues in $\Lambda_1$, and matrices $X_{R1}$ and $X_{R2}$ are $n \times r$ real matrices of eigenvectors corresponding to the real eigenvalues in $U_2$ and $U_3$, respectively. Note that the structure of $X$ is consistent with that of $J$ in (1).

The following theorem establishes a one-to-one connection between Jordan pairs constructed in this way and matrices $V$ which define SPS transformations of $C_0$ as defined in Definition 2.

THEOREM 3. [illegible] $J$ [illegible] (1) [illegible] $\det(U_2 - U_3) \neq 0$

(a) [illegible] $X$ [illegible] (10) [illegible] $(X, J)$ [illegible] $Z$ [illegible] (5)

$$V = \left[\begin{array}{c} X \\ XJ \end{array}\right] Z \tag{11}$$

[illegible] $C_0$

(b) [illegible] $V$ [illegible] $C_0$ [illegible] $Z$ [illegible] (5) [illegible] $X$ [illegible] (10) [illegible] (11) [illegible] $(X, J)$ [illegible]

[illegible] By definition of a Jordan pair and using Lemma 1, it is found that $V$ of (11) is nonsingular. Then compute with block matrices to find

$$V_{21} = \left[\begin{array}{cc} (X_c + \overline{X_c})\Omega_1^2 & -(X_{R1} + X_{R2})U_2U_3 \end{array}\right],$$

$$V_{12} = \left[\begin{array}{cc} -(X_c + \overline{X_c}) & X_{R1} + X_{R2} \end{array}\right],$$

and it can be checked that $V_{21} = -V_{12}\Omega^2$. Similarly, it is found that

$$V_{11} = \left[\begin{array}{cc} X_c\overline{\Lambda} + \overline{X_c}\Lambda & -X_{R1}U_3 - X_{R2}U_2 \end{array}\right],$$

$$V_{22} = \left[\begin{array}{cc} -(X_c\Lambda + \overline{X_c\Lambda}) & X_{R1}U_2 + X_{R2}U_3 \end{array}\right],$$

and, finally, that $V_{22} = V_{11} + 2V_{12}U$. Now part (a) follows from Lemma 2, provided that $V$ is a real matrix.

However, using (5), it follows that

$$\begin{aligned} V &= \left[\begin{array}{cccc} X_c & X_{R1} & X_{R2} & \overline{X_c} \\ X_c\Lambda_1 & X_{R1}U_2 & X_{R2}U_3 & \overline{X_c\Lambda} \end{array}\right] Z \\ &= \left[\begin{array}{cccc} X_c\overline{\Lambda} + \overline{X_c}\Lambda & -X_{R1}U_3 - X_{R2}U_2 & X_{R1} + X_{R2} & -(X_c + \overline{X_c}) \\ (X_c + \overline{X_c})\Omega_1^2 & -(X_{R1} + X_{R2})U_2U_3 & X_{R1}U_2 + X_{R2}U_3 & -(X_c\Lambda_1 + \overline{X_c\Lambda_1}) \end{array}\right] \end{aligned} \tag{12}$$

and is clearly a real matrix (as Definition 2 requires).

For the converse, observe first that, under condition (6), $C_0Z^{-1} = Z^{-1}J$, and it follows from this equation that the columns of $Z^{-1}$ are right eigenvectors of $C_0$. If $V$ defines an SPS of $C_0$, then using the defining equation (8),

$$C = VZ^{-1}J(VZ^{-1})^{-1}. \tag{13}$$

Thus, the columns of $VZ^{-1}$ are eigenvectors of $C$. Since $C$ has the same spectrum as $C_0$ (and $J$), this matrix of eigenvectors can be written in the form

$$VZ^{-1} = \begin{bmatrix} X \\ XJ \end{bmatrix} = \begin{bmatrix} X_c & X_{R1} & X_{R2} & \overline{X_c} \\ X_c\Lambda & X_{R1}U_2 & X_{R2}U_3 & \overline{X_c\Lambda} \end{bmatrix}. \tag{14}$$

Thus, $X$ has the required form and, since $VZ^{-1}$ is nonsingular, $(X,\, J)$ form a Jordan pair. □

**5. Generating real isospectral systems.** Computational procedures for generating isospectral families of real systems can be formulated from the preceding analysis. This is done first in the language of SPS, and then in terms of Jordan pairs.

1. Fix the diagonal matrix of eigenvalues, $J$, with the form (1). Form matrices $Z$ of (5) and $C_0$ of (4).
2. Assign the $n \times 2n$ matrix of eigenvectors, $X$ with the form (10), in such a way that $(X,\, J)$ form a Jordan pair. (Clearly, this can be done in many ways.)
3. Compute $V = [{X \atop XJ}]Z$.
4. Compute $C = VC_0V^{-1}$ and read off the submatrices $M^{-1}K = -C_{21}$ and $M^{-1}D = -C_{22}$.
5. Assign a nonsingular real mass matrix $M$ and compute $K = -MC_{21}$, $D = -MC_{22}$.

The alternative procedure is based on the notion of a Jordan triple. Thus, given the Jordan pair of item 2 above, [illegible] a real nonsingular mass matrix $M$ and determine a $2n \times n$ matrix $Y$ satisfying

$$\begin{bmatrix} X \\ XJ \end{bmatrix} Y = \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}, \tag{15}$$

and $(X,\, J,\, Y)$ is known as a [illegible].

When $Y$ has been determined, the [illegible]

$$\Gamma_j = XJ^jY, \qquad j = 0, 1, 2, 3, \tag{16}$$

can be formed, and the system coefficients can be defined recursively in terms of the moments (see Theorem 2 of [9], for example):

$$M = \Gamma_1^{-1}, \qquad D = -M\Gamma_2M, \qquad K = -M\Gamma_3M + D\Gamma_1D. \tag{17}$$

The alternative procedure for generating an isospectral family of real systems is now as follows:

1. Fix the diagonal matrix of eigenvalues, $J$, with the form (1).
2. Assign the $n \times 2n$ matrix of eigenvectors, $X$ with the form (10), in such a way that $(X,\, J)$ form a Jordan pair. (Clearly, this can be done in many ways.)
3. Assign a nonsingular real mass matrix $M$ and solve (15) for $Y$.
4. Compute the moments (16) and hence the coefficients $D$ and $K$ from (17).

Consider first an obvious construction of isospectral systems. Given a set of eigenvalues with self-conjugate symmetry (as in (1)), the diagonal system $L_0(\lambda)$ of (2) has this spectrum. Then a class of real isospectral systems is obtained by applying a real strict equivalence transformation to $L_0(\lambda)$. Indeed, a class of real symmetric (resp., Hermitian) systems can be generated by applying a real (resp., complex) congruence. However, it is easily seen that with any of these constructions, each pair of eigenvalues determined by a diagonal entry of $L_0(\lambda)$ has a common eigenspace of dimension at least one. This property is generally unnatural for systems of physical origin, and our interest is in more general constructions.

3. We will construct a $4 \times 4$ real system with 4 real eigenvalues and 4 nonreal eigenvalues. Take a Jordan matrix of the form (1) with blocks

$$\Lambda = \operatorname{diag}[-1+i,\ -4+i], \quad U_2 = \operatorname{diag}[-0.5,\ -1], \quad U_3 = \operatorname{diag}[-3, -4].$$

Then take a matrix $X$ of the form (10) with blocks

$$X_c = \begin{bmatrix} 0.0625(1-i) & (0.6)(1-0.1i) \\ 0.2500(1-i) & (0.6)(1-0.1i) \\ 0.5625(1-i) & 0 \\ 1.0000(1-i) & (-1)(1-0.1i) \end{bmatrix},$$

$$X_{R1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad X_{R2} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix}.$$

It is found that this data generate the real monic system with

$$D = \begin{bmatrix} -6.0008 & 6.5981 & 5.8527 & -4.4416 \\ -8.4540 & 8.7557 & 6.1483 & -4.6190 \\ 7.2668 & -8.8863 & 6.8695 & -0.9717 \\ 22.9877 & -21.5283 & -5.2094 & 8.8756 \end{bmatrix},$$

$$K = \begin{bmatrix} -20.7791 & 15.4356 & 19.5039 & -13.4062 \\ -22.8678 & 16.8592 & 20.4793 & -13.8052 \\ 1.2225 & -7.0538 & 11.2192 & -3.4988 \\ 35.3128 & -32.9667 & -18.4409 & 18.4076 \end{bmatrix}.$$

These calculations can then be checked by showing that the eigenvalues of this monic system are, indeed, those specified in $J$.

**6. Symmetric systems, part 1.** The next objective is, of course, to determine the matrices $V$ defining an SPS of $C_0$, and which also generate systems. The question of when these coefficients satisfy positivity conditions will be considered later.

At this point it is necessary to introduce the rather subtle notion of the [2] associated with the real eigenvalues. (The reader is referred to the references listed in the footnote for formal definitions, but for the uninitiated, Appendix B gives an intuitive introduction to this important notion.) For systems with symmetries it is not enough to allocate arbitrary real eigenvalues; the invariants of the sign characteristic must also be specified. With our hypotheses on the spectrum, this can be accomplished by introducing the matrix

$$P = \begin{bmatrix} 0 & 0 & 0 & I_{n-r} \\ 0 & I_r & 0 & 0 \\ 0 & 0 & -I_r & 0 \\ I_{n-r} & 0 & 0 & 0 \end{bmatrix}. \tag{18}$$

---

[2] See [3], [4], [9], and the expository Appendix B to this paper.

Notice that, with $J$ of the form (1), $(PJ)^* = PJ$. Thus, although $J$ is generally not Hermitian, $PJ$ is so.

Symmetry of the coefficients of the system follows if a symmetric $M$ is chosen and, also, if $Y = PX^*$ is the only solution of (15). For then the $\Gamma$'s are Hermitian and, using (17), so are $D$ and $K$ (and when this is the case, $(X, J, PX^*)$ is said to be a [illegible]). If, in addition, $X$ has the block structure of (10), then the moments and the system coefficients will be real and symmetric.

Thus, if a self-adjoint triple is to be constructed, then (see (15)) $XY = XPX^* = 0$. Thus, once admissible matrices $J$ and $P$ have been assigned, the crux of the problem is to find an $X$ such that $XPX^* = 0$ and $X(PJ)X^*$ is nonsingular. In [9] a geometric approach is taken for the determination of such matrices $X$. Here, attention is focused on [illegible] systems so that the structure of (10) is also to be imposed on $X$. In this case, $XPX^* = 0$ can be written in the form

$$X_c X_c^T + \overline{X_c X_c^T} = -X_{R1} X_{R1}^T + X_{R2} X_{R2}^T. \tag{19}$$

Now this equation simply says that the real part of the matrix $X_c X_c^T$ takes the value $\frac{1}{2}(-X_{R1} X_{R1}^T + X_{R2} X_{R2}^T)$ and does not constrain the imaginary part.

Consequently, it follows from (19) that

$$X_c X_c^T = R_1 - iR, \tag{20}$$

where

$$R_1 := \frac{1}{2}(-X_{R1} X_{R1}^T + X_{R2} X_{R2}^T) = \begin{bmatrix} X_{R1} & X_{R2} \end{bmatrix} \begin{bmatrix} -I_r & 0 \\ 0 & I_r \end{bmatrix} \begin{bmatrix} X_{R1}^T \\ X_{R2}^T \end{bmatrix} \tag{21}$$

and $R$ is a real symmetric matrix.

Notice also that, if $R_1 - iR$ is designed to have rank $n - r$, then $\operatorname{rank}(X_c) \geq n - r$. (To see that equality need not be the case, consider the product $A_0 A_0^T$ where $A_0 = \begin{bmatrix} 1 & i \\ i & -1 \end{bmatrix}$.)

The broad strategy suggested here is to assign all the real eigenvectors, and hence the matrix $R_1$. Generically, it can be expected that the eigenvectors associated with the eigenvalues in the upper half-plane will be linearly independent. Thus, the matrix $X_c \in \mathbb{C}^{n \times (n-r)}$ of (10) will have full rank, $n - r$. Now there is a standard method for finding a symmetric factorization of a complex symmetrix matrix (as required in (20)), in which the rank of the factors is equal to that of the given right-hand side. So the problem reduces to the following: Given $R_1$, with rank determined by the choice of real eigenvectors, find an $R$ such that $R_1 - iR$ has rank $n - r$.

The "standard method" for symmetric factorization referred to above was developed by Takagi in the 1920s. A quick introduction, based on the exposition and algorithm of [1], is given in Appendix A of this paper.

It is instructive to consider a simple example at this stage.

[illegible] 4. We construct a $2 \times 2$ system with two real eigenvalues and one complex pair. The spectral data are

$$J = \begin{bmatrix} -2+i & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -2-i \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

(a) We first prescribe the [illegible] eigenvectors:

$$X_{R1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad X_{R2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then calculate to find

$$R_1 = \frac{1}{2}(-X_{R1}X_{R1}^T + X_{R2}X_{R2}^T) = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}.$$

Choosing

$$R = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$$

yields the rank one matrix

$$R_1 - iR = \frac{1}{2}\begin{bmatrix} -1 & -i \\ -i & 1 \end{bmatrix},$$

and this has the factorization (cf. (20)) $R_1 - iR = X_c X_c^T$, where

$$X_c = \frac{1}{\sqrt{2}}\begin{bmatrix} -i \\ 1 \end{bmatrix}.$$

Now compute with (16) and (17) to find the real symmetric system

$$M = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 3 \\ 3 & -5 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 2 \\ 2 & -6 \end{bmatrix},$$

and it can be checked that the spectrum is, indeed, that prescribed.

(b) In contrast to (a), if $X_{R1} = X_{R2} = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$, then $R_1 = 0$ and $R$ is chosen so that $R_1 - iR$ has rank one, say $R = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$, so that $R_1 - iR = \begin{bmatrix} 0 & 0 \\ 0 & i \end{bmatrix}$, in which case we take

$$X_c = \begin{bmatrix} 0 \\ e^{i\pi/4} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}}(1+i) \end{bmatrix}$$

and

$$X = \begin{bmatrix} 0 & 1 & 1 & 0 \\ \frac{1}{\sqrt{2}}(1+i) & 0 & 0 & \frac{1}{\sqrt{2}}(1-i) \end{bmatrix}.$$

Following the steps above, it is found that this determines the [illegible] system,

$$M = \begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}, \quad D = \begin{bmatrix} 3 & 0 \\ 0 & -2 \end{bmatrix}, \quad K = \begin{bmatrix} 2 & 0 \\ 0 & \frac{-5}{2} \end{bmatrix}.$$

The fact that the system is diagonal can be attributed to the choice of linearly dependent real eigenvectors (see the discussion before Example 3).

An interesting feature of Example 4 is the fact that $R$ is first chosen to [illegible] the rank of $R_1 - iR$ relative to that of $R_1$. Clearly, this was necessary, as $X_c$ must have rank one. In the second case, $R$ is chosen to augment the rank of $R_1$. Features of this kind are a major difficulty in the design of a general strategy for finding solutions to (19). Notice also that the necessary condition that $X$ have full rank was guaranteed by our choice of the real eigenvectors the first time around, but not the second.

Another approach to the determination of solutions of (19) begins with assigning a [illegible] matrix of eigenvectors $X_c$ and then solving (19) for real matrices $X_{R1}$ and $X_{R2}$. This line of attack is postponed to a future investigation.

**7. Symmetric systems, part 2.** The problem of finding matrices $X$ of the form (10) which also satisfy $XPX^* = 0$ has been reformulated in the form of (20), where $R_1$ is given by (21). This matrix is to be assigned, and then a real symmetric matrix $R$ is to be chosen in such a way that $R_1 - iR$ has rank $n - r$. To ensure that both real and nonreal eigenvalues appear, it is assumed that $1 \leq r \leq n - 1$.

It appears that the rank of $R_1$ can take any value between zero (when $X_{R2} = X_{R1}$) and $n$ (when the real eigenvectors span the whole space). A complete understanding of our problem seems to require knowledge of the connections between

$$\operatorname{rank}(R_1), \quad \operatorname{rank}(R), \quad \text{and} \quad \operatorname{rank}(R_1 - iR).$$

As this seems not to be well known, the details are provided in Appendix C. In particular, Theorem 9 shows that, to achieve

$$n - r = \operatorname{rank}(R_1 - iR) < \operatorname{rank}(R_1),$$

which can certainly be physically reasonable, $R$ must be chosen so that $\pm i$ become eigenvalues of the real symmetric pencil $R_1 - \lambda R$. Now this phenomenon arose in Example 4, apparently fortuitously! But, in fact, Theorem 9 shows that this choice of $R$ was essentially unique.

More generally, notice that although the factor

$$\begin{bmatrix} -I_r & 0 \\ 0 & I_r \end{bmatrix}$$

in (21) has rank $2r$, the "modified" matrix

$$\begin{bmatrix} -I_r & iI_r \\ iI_r & I_r \end{bmatrix}$$

has rank $r$ with eigenvalues $\pm i$ repeated $r$ times. Thus, by choosing

$$R = \begin{bmatrix} X_{R1} & X_{R2} \end{bmatrix} \begin{bmatrix} 0 & I_r \\ I_r & 0 \end{bmatrix} \begin{bmatrix} X_{R1}^T \\ X_{R2}^T \end{bmatrix}, \tag{22}$$

$$R_1 - iR = \begin{bmatrix} X_{R1} & X_{R2} \end{bmatrix} \begin{bmatrix} -I_r & iI_r \\ iI_r & I_r \end{bmatrix} \begin{bmatrix} X_{R1}^T \\ X_{R2}^T \end{bmatrix},$$

we introduce eigenvalues $\pm i$ of multiplicity $r$ into the pencil $R_1 - iR$. Modifications of this definition for $R$ are easily devised to generate real symmetric matrices $R_1 - iR$ with rank $\rho$, where $r \leq \rho \leq 2r$.

If $R_1$ has rank $n - r$, then by choosing an $R$ with the same range as $R_1$, an $R_1 + iR$ can be constructed with the same rank, $n - r$. However, with this construction, the necessary condition that the range of $X$ have dimension $n$ cannot be satisfied. Indeed, there seems to be a difficult problem here when the rank of $R_1$ is low. There may be a deep property that is not fully understood to the effect that, although linear dependencies among the real eigenvectors are known to be possible, the dimension of the span of the real eigenvectors cannot be "too low." The parameters used in Theorem 9 of Appendix C will probably play a role in any resolution of this problem. Indeed, the sets of admissible parameters can be analyzed using the canonical forms found in Theorem 9.2 of [11] and described in Appendix C.

These techniques are not investigated more deeply here, and we conclude the present discussion with a numerical illustration.

5. The program here is to take the data from Example 3, which were used in the design of a real (nonsymmetric) system, augment it with a sign characteristic, and design a real system. Thus, as in Example 3, we take $n = 4$, $r = 2$, and

$$\Lambda = \operatorname{diag}[-1+i,\ -4+i], \quad U_2 = \operatorname{diag}[-0.5,\ -1], \quad U_3 = \operatorname{diag}[-3, -4].$$

Now consider (20). Since there are two pairs of nonreal eigenvalues, $X_c$ is to be constructed with rank 2. Following the strategy leading to (22) results in

$$A = R_1 + iR = \begin{bmatrix} 2i & 1+i & -1+i & -1-i \\ & 0 & 0 & -2 \\ & & 0 & -2i \\ & & & 0 \end{bmatrix}.$$

Notice that $R_1$ is defined by the data of Example 3 and $R$ is chosen as in (22). Using the Takagi algorithm, we obtain

$$X_c = \begin{bmatrix} -1.0082(1+i) & 0.1281(-1+i) \\ -0.8801 & (-0.8801)i \\ (-0.8801)i & 0.8801 \\ 1.1362 & (-1.1362)i \end{bmatrix},$$

and construction of the $4 \times 8$ matrix $X$ is complete.

The Jordan matrix is now $\operatorname{diag}[-1+i,\ -4+i,\ -0.5,\ -1,\ -3,\ -4,\ -1-i,\ -4-i]$, and (see (18))

$$P = \begin{bmatrix} 0 & 0 & 0 & I_2 \\ 0 & I_2 & 0 & 0 \\ 0 & 0 & -I_2 & 0 \\ I_2 & 0 & 0 & 0 \end{bmatrix}.$$

It can be verified that $XPX^* = 0$, and the formulae of (16) and (17) are applied to produce the real symmetric system:

$$M = \begin{bmatrix} 0.4496 & -0.3267 & 1.6481 & -0.2840 \\ & 0.2748 & -0.6290 & 0.0519 \\ & & 0.8991 & 0.2533 \\ & & & -0.0696 \end{bmatrix},$$

$$D = \begin{bmatrix} -8.4914 & 4.2104 & 4.9463 & -4.5620 \\ & -1.6488 & -0.6350 & 1.7465 \\ & & 6.5591 & -1.7314 \\ & & & -0.3876 \end{bmatrix},$$

$$K = \begin{bmatrix} 0.4612 & -3.0335 & -4.5721 & 4.8244 \\ & 3.8425 & 6.4105 & -5.2257 \\ & & 12.0600 & -9.8717 \\ & & & 7.8178 \end{bmatrix}.$$

As usual, it can be verified that this system has the spectrum determined by $J$. It is interesting that, in spite of the location of all the eigenvalues in the left half-plane, [illegible] coefficients are indefinite.

Of course, the analysis simplifies if there are to be [illegible] real eigenvalues (as in [10]), for then $X_c$ is nonsingular and it is necessary only to assign a nonsingular real symmetric $R$. On the other hand, if $X_c$ is a solution of (19), so is $X_c\Theta$ for any real orthogonal matrix $\Theta$. Thus, for a fixed right-hand side of (19), a family of solutions $X$ is obtained depending on $\frac{1}{2}(n-r)(n-r-1)$ real parameters. This is consistent with results obtained in [10] for the case $r = 0$.

The situation in which there are no [illegible] eigenvalues is also of great interest and includes the so-called overdamped, hyperbolic, and quasi-hyperbolic systems. They are the topics of section 10 below.

**8. Positivity of $M$, $D$, and $K$.** In this section it is assumed that the spectral data are consistent with real and symmetric systems, and we examine the further conditions required to ensure positive definite (or possibly semidefinite) coefficients $M$, $D$, $K$. It will be convenient to make a further simplifying hypothesis, namely, that systems are to be designed which are [illegible]. This is equivalent to the hypothesis that $K$ is nonsingular. This can be justified here on the grounds that, in this section, our major interest is in [illegible] systems, i.e., those with all eigenvalues in the open left half of the complex plane. In other words, $J$ of (1) is to be a stable matrix.

In this case there is a nice alternative to the formula

$$K = -M\Gamma_3 M + D\Gamma_1 D$$

of (17) (see Theorem 2 of [10], for example). Thus, given a self-adjoint triple ($X$, $J$, $PX^*$), we have

$$\Gamma_{-1} := X(J^{-1}P)X^* = -K^{-1}. \tag{23}$$

This follows immediately from the [illegible] for $L(\lambda)$ expressed here in terms of any Jordan triple:

$$L(\lambda)^{-1} = X(\lambda I_{2n} - J)^{-1}Y.$$

LEMMA 4. [illegible] $J$ [illegible] $\Gamma_1$ [illegible] $M$, $D$, $K$ [illegible] $\Gamma_1$ $-\Gamma_2$ [illegible] $-\Gamma_{-1}$ [illegible]

[illegible]. Observe that, if 0 is not an eigenvalue of $J$, then $\Gamma_{-1}$ is well defined, and the lemma follows from the first two relations of (17) together with (23). □

Since the moments are readily computed from a Jordan triple, this immediately suggests that the positivity of $M$, $D$, $K$ could be checked by trial and error. However, a more precise result can be proved, which generalizes Theorem 9 of [10]. Notice the important role played by positivity of the [illegible] moment in this result.

THEOREM 5. [illegible] $J$ [illegible] $\Gamma_2 \leq 0$ [illegible] $\Gamma_1, \Gamma_{-1}$ [illegible] $M > 0$ $D \geq 0$, [illegible] $K > 0$

[illegible]. Since $\Gamma_1$ and $\Gamma_{-1}$ are nonsingular, $M$ and $K$ are well defined by (17) and (23). Then the stability of $J$, together with Theorem 7 of [13], implies that $M > 0$ and $K > 0$. Then $D \geq 0$ follows from $\Gamma_2 \leq 0$ and (17). □

Note that there is, of course, a classical converse statement for Theorem 5: If $M > 0$, $D \geq 0$, and $K > 0$, then all eigenvalues are in the (possibly closed) left half-plane.

In general, it will be difficult to apply the last result numerically ab initio. A major open question (to which partial contributions are made here) is the following.

[illegible] 1. Given that the spectrum is stable, what further conditions on $X$ (the matrix of [illegible]) will ensure that the coefficients of the system are positive definite?

A closely connected question arising in section 7 is as follows.

[illegible] 2. What are the constraints linking the dimensions of the ranges of the matrices $\begin{bmatrix} X_{R1} & X_{R2} \end{bmatrix}$ and $\begin{bmatrix} X_c & \overline{X_c} \end{bmatrix}$?

[illegible] 6. Reconsider Example 4. Modify the data and take $X_{R1} = X_{R2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, and retain the same matrices $J$ and $P$. Now $R_1 = 0$ and we choose $R = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. It is found that a diagonal system with positive definite coefficients is generated:

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad D = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}, \quad K = \begin{bmatrix} 2 & 0 \\ 0 & 2.5 \end{bmatrix}.$$

**9. Hyperbolic and overdamped systems.** For the purpose of this discussion, consider systems with a positive definite leading coefficient, $M$. It is well known that there are symmetric vibrating systems of practical interest for which all eigenvalues are real. An early and influential paper on this subject was that of Duffin [2] concerning overdamped systems. Subsequently, it has been realized that systems with real spectrum arise more generally. Thus, systems with all $2n$ eigenvalues real with $n$ of positive type (with a $+1$ in the sign characteristic) and $n$ of negative type (with a $-1$ in the sign characteristic) are said to be [illegible]. If, furthermore, the sets of eigenvalues of the two types are separated (all positive-type eigenvalues greater than all negative-type eigenvalues), then the system is [illegible]. Finally, a system which is hyperbolic and in which $D > 0$, $K \geq 0$ is said to be [illegible]. Given these inequalities it is equivalent to say that a hyperbolic system is overdamped if all eigenvalues are negative. (Numerical methods for such problems are the topic of [7] and [6], for example.)

The distinction between these classes of systems is quite clear in our context of inverse problems. Let us begin with hyperbolic systems. Thus, all eigenvalues are real, and the data for the inverse problem consist of diagonal matrices

$$J = \begin{bmatrix} U_2 & 0 \\ 0 & U_3 \end{bmatrix}, \quad P = \begin{bmatrix} I_n & 0 \\ 0 & -I_n \end{bmatrix} \tag{24}$$

(cf. (1) and (18)). Furthermore, the smallest eigenvalue of $U_2$ (having positive type) exceeds the largest eigenvalue of $U_3$ (having negative type). Thus,

$$\max_{u_j \in U_3}(u_j) < \min_{u_k \in U_2}(u_k). \tag{25}$$

Then the necessary condition (6) of Lemma 1 is satisfied.

For the more general quasi-hyperbolic systems, $U_2$ and $U_3$ are simply specified in such a way that (6) holds but (25) does not necessarily hold.

Now an isospectral family of quasi-hyperbolic systems with spectrum defined by $U_2$ and $U_3$ is determined by full rank eigenvector matrices

$$X = \begin{bmatrix} X_{R1} & X_{R2} \end{bmatrix},$$

where $X_{R1}, X_{R2} \in \mathbb{R}^{n \times n}$ and

$$XPX^* = X_{R1}X_{R1}^T - X_{R2}X_{R2}^T = 0$$

(cf. (10), (19)). This condition is easily satisfied: Since $X$ has full rank, both $X_{R1}$ and $X_{R2}$ must be nonsingular, so we may take an arbitrary $A > 0$ in $\mathbb{R}^{n\times n}$ and then choose $X_{R1}$ and $X_{R2}$ so that

$$X_{R1}X_{R1}^T = X_{R2}X_{R2}^T = A. \tag{26}$$

Natural choices for $X_{R1}$ and $X_{R2}$ are then $A^{1/2}$, or a lower triangular matrix generated by a Cholesky factorization of $A$ (see [5], for example). Having made a first choice of $X_{R1}$ and $X_{R2}$, infinitely many more candidates are generated by multiplying on the right with a real orthogonal matrix. In particular, once a nonsingular $X_{R1}$ is chosen, one may take

$$X_{R2} = X_{R1}\Theta,$$

where $\Theta$ is real orthogonal. We adopt this strategy.

Then it is easily verified that the following formulae hold: For the moments,

$$\begin{aligned} \Gamma_1 &= X_{R1}(U_2 - \Theta U_3\Theta^T)X_{R1}^T, \\ \Gamma_2 &= X_{R1}(U_2^2 - \Theta U_3^2\Theta^T)X_{R1}^T, \\ \Gamma_{-1} &= X_{R1}(U_2^{-1} - \Theta U_3^{-1}\Theta^T)X_{R1}^T, \end{aligned} \tag{27}$$

and for the coefficients,

$$\begin{aligned} M &= X_{R1}^{-T}(U_2 - \Theta U_3\Theta^T)^{-1}X_{R1}^{-1}, \\ D &= -X_{R1}^{-T}(U_2 - \Theta U_3\Theta^T)^{-1}(U_2^2 - \Theta U_3^2\Theta^T)(U_2 - \Theta U_3\Theta^T)^{-1}X_{R1}^{-1}, \\ K &= -X_{R1}^{-T}(U_2^{-1} - \Theta U_3^{-1}\Theta^T)^{-1}X_{R1}^{-1}. \end{aligned} \tag{28}$$

It is immediately apparent that $X_{R1}$ merely determines a simultaneous congruence applied to the three system coefficients. Once the spectrum is specified in the form of $U_2$ and $U_3$, the coefficients are determined (to within this simultaneous congruence) by the choice of $\Theta$. Thus, the [illegible] of $M$, $D$, $K$ do not depend on $X_{R1}$. (A similar phenomenon arises in the case when all eigenvalues are [illegible]; see (34)–(36) and Theorem 13 of [10].)

Theorem 5 now provides criteria for generating families of real hyperbolic systems.

Corollary 6. [illegible] $\Lambda$ [illegible] $W$ [illegible] (1) [illegible] $U_2 < 0$ $U_3 < 0$ [illegible] $\det(U_2 - U_3) \neq 0$ [illegible]

(a) $X_{R1} \in R^{n\times n}$ [illegible]

(b) $\Theta$ [illegible]

$$U_2^2 \leq \Theta U_3^2\Theta^T \tag{29}$$

[illegible] $U_2 - \Theta U_3\Theta^T$ $U_2^{-1} - \Theta U_3^{-1}\Theta^T$ [illegible]

[illegible] (28)) $M > 0$ $D \geq 0$ $K > 0$ [illegible] $M\lambda^2 + D\lambda + K$ [illegible]

[illegible] (25) [illegible]

[illegible]. Condition (29) and the equation for $\Gamma_2$ in (27) ensure that $\Gamma_2 \geq 0$. Also, from (27), $\Gamma_1$ and $\Gamma_{-1}$ are nonsingular. So the result follows from Theorem 5. ☐

7. Clearly, given the hypotheses of the corollary on $U_2$ and $U_3$, $X_{R1} = I_n$ and $\Theta = I_n$ are admissible choices. If we write

$$U_2 = \operatorname{diag}[\mu_1^{(2)}, \mu_2^{(2)}, \dots, \mu_n^{(2)}], \quad U_3 = \operatorname{diag}[\mu_1^{(3)}, \mu_2^{(3)}, \dots, \mu_n^{(3)}],$$

(28) determines the diagonal system with diagonal entries

$$\frac{\lambda^2 - (\mu_j^{(2)} + \mu_j^{(3)})\lambda + \mu_j^{(2)}\mu_j^{(3)}}{\mu_j^{(2)} - \mu_j^{(3)}} = \frac{(\lambda - \mu_j^{(2)})(\lambda - \mu_j^{(3)})}{\mu_j^{(2)} - \mu_j^{(3)}}.$$

8. Take $X_{R1} = I_4$ and

$$U_2 = \operatorname{diag}[-1,\ -2,\ -3,\ -4], \qquad U_3 = \operatorname{diag}[-5,\ -6,\ -7,\ -8].$$

Consider the orthogonal matrix

$$\Theta = \frac{1}{10}\begin{bmatrix} 2 & -8 & 4 & -4 \\ 8 & -2 & -4 & 4 \\ 4 & 4 & -3 & -8 \\ 4 & 4 & 8 & 2 \end{bmatrix}$$

and verify that the hypotheses of the corollary are satisfied. Apply the formulae of (28) to generate the overdamped system (with truncated decimal form)

$$M = \begin{bmatrix} 0.1886 & 0.0269 & -0.0168 & -0.0051 \\ & 0.2896 & 0.0690 & 0.0707 \\ & & 0.2694 & 0.0808 \\ & & & 0.42342 \end{bmatrix},$$

$$D = \begin{bmatrix} 1.3771 & 0.0808 & -0.0673 & -0.0253 \\ & 2.1582 & 0.3451 & 0.4242 \\ & & 2.6162 & 0.5657 \\ & & & 4.3939 \end{bmatrix},$$

$$K = \begin{bmatrix} 1.1886 & 0.0539 & -0.0505 & -0.0202 \\ & 3.1582 & 0.4141 & 0.5657 \\ & & 5.4242 & 0.9697 \\ & & & 10.7879 \end{bmatrix},$$

with eigenvalues $-1$, $-2$, $-3$, $-4$ of positive type, and $-5$, $-6$, $-7$, $-8$ of negative type. An associated matrix of eigenvectors has the form

$$X = [\ I \quad \Theta\ ] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0.2 & -0.8 & 0.4 & -0.4 \\ 0 & 1 & 0 & 0 & 0.8 & -0.2 & -0.4 & 0.4 \\ 0 & 0 & 1 & 0 & 0.4 & 0.4 & -0.2 & -0.8 \\ 0 & 0 & 0 & 1 & 0.4 & 0.4 & 0.8 & 0.2 \end{bmatrix}.$$

**10. Conclusions.** Real damped vibrating systems defined by $n \times n$ coefficient matrices $M$, $D$, and $K$ have been studied, with the simplifying hypothesis of semi-simple Jordan structure, i.e., associated $2n \times 2n$ Jordan canonical forms, $J$, are diagonal. A corresponding primitive companion matrix, $C_0$, is formulated (equation (4)) and plays a significant role. The equivalence between "structure preserving similarities" of $C_0$ and Jordan pairs for the system has been established in Theorem 3.

These constructions have been used to find solutions to the following inverse problem: Given $J$, find consistent real vibrating systems (see section 5). The corresponding problem for consistent [illegible] real systems (with mixed real and nonreal spectrum) is more complicated. A partial solution for this problem (taking advantage of Takagi's factorization of symmetric complex matrices) is the subject of sections 6 and 7. To ensure that coefficient matrices $M$, $D$, $K$ have definiteness properties is more difficult again, but significant insight is provided in section 8 in terms of "moments" of the system.

Similar methods applied to systems having [illegible] real eigenvalues are more tractable (and include systems of hyperbolic and overdamped types). They are developed in section 9, where parametrizations of isospectral systems by real orthogonal matrices arise naturally. This compares nicely with similar results of [10] for systems at the other (elliptic) extreme with [illegible] real eigenvalues.

The analysis developed here requires knowledge of the relationship between the ranks of a complex symmetric matrix and its real and imaginary parts. This is clarified in Theorem 9 of Appendix C. The result seems to be new and may be of more general interest.

**Appendix A. Takagi's factorization.** A method for making the factorization needed in sections 6 and 7 is attributed to Takagi and dates from the 1920s. For a given complex symmetric matrix $A \in \mathbb{C}^{n\times n}$ of rank $n-r$, a factorization $A = X_c X_c^T$ is produced in which $X_c$ also has rank $n-r$. Computer programs are now available for this task (see Bunse-Gerstner and Gragg [1]). There is also a careful discussion of this in section 4.4.4 of Horn and Johnson [8]. Here, an introduction is made to the relatively simple case in which nonzero singular values of the right-hand side of (20) (i.e., when $A = R_1 - iR$) are distinct. It is based on the presentation of [1].

1. Let $A$ denote the (given) complex symmetric right-hand side of (20) and assume that $\operatorname{rank}(A) = n-r$. Form the singular value decomposition $A = U\Sigma V^*$, where $U$ and $V$ are unitary matrices and
$$\Sigma = \operatorname{diag}\begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_n \end{bmatrix}$$
with $\sigma_1 > \sigma_2 > \cdots > \sigma_{n-r} > 0$ and $\sigma_{n-r+1} = \ldots = \sigma_n = 0$ (see [5] for further details).
2. Let $u_j$ and $v_j$ denote the columns of $U$ and $V$, respectively, and compute $q_j^2 := u_j^T v_j$, $j = 1, 2, \ldots, n-r$ (note that $q_j^2$ will generally be complex).
3. Form $\Sigma_1 := \operatorname{diag}\begin{bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_{n-r} \end{bmatrix}$ of size $(n-r)\times(n-r)$ and form the $n\times(n-r)$ matrix $U_0 = \begin{bmatrix} u_1 & u_2 & \cdots & u_{n-r} \end{bmatrix}$.
4. Compute a matrix $Q = \operatorname{diag}\begin{bmatrix} q_1 & q_2 & \cdots & q_{n-r} \end{bmatrix}$.
5. Compute $X_c = U_0 \bar{Q} \Sigma_1^{1/2}$ (of size $n\times(n-r)$).

Let us quickly confirm that this produces the required symmetric factorization. Since $A$ is symmetric,

$$A = U\Sigma V^* = \overline{V}\Sigma U^T.$$

But the singular vectors for the (distinct) nonzero singular values are unique to within a scalar multiplier of modulus one. Thus, there are numbers $\omega_1, \ldots, \omega_{n-r}$ such that

$$v_j = \omega_j \overline{u_j}, \quad |\omega_j| = 1, \quad j = 1, 2, \ldots, n-r.$$

Defining $U_0$ as in item 3, $V_0 = \begin{bmatrix} v_1 & \cdots & v_{n-r} \end{bmatrix}$ and $\Omega = \operatorname{diag}\begin{bmatrix} \omega_1 & \ldots & \omega_{n-r} \end{bmatrix}$, we have $V_0 = \overline{U_0}\Omega$, or

$$V_0^* = \overline{\Omega} U_0^T.$$

Furthermore, $u_j^T v_j = \omega_j (u_j^T \overline{u_j}) = \omega_j$ for $j = 1, 2, \ldots, n-r$, so that (see item 4) $\Omega = Q^2$.

Now compute

$$\begin{aligned} X_c X_c^T &= (U_0 \overline{Q} \Sigma^{\frac{1}{2}})(\Sigma^{\frac{1}{2}} \overline{Q} U_0^T) \\ &= U_0 \Sigma \overline{Q}^2 U_0^T = U_0 \Sigma (\overline{\Omega} U_0^T) \\ &= U_0 \Sigma V_0^* = U \Sigma V^* = A. \end{aligned}$$

The purpose of the next example is simply to illustrate this scheme. Calculations are completed in MATLAB.

[illegible] 9. Let $A = \begin{bmatrix} i & i \\ i & 0 \end{bmatrix}$ and note that $n = 2$, $r = 0$ (so no singular values are equal to zero). The MATLAB singular value decomposition yields (with truncated numbers)

$$U = \begin{bmatrix} -0.8507 & -0.5257 \\ -0.5257 & 0.8507 \end{bmatrix} i, \qquad V = \begin{bmatrix} -0.8507 & 0.5257 \\ -0.5257 & -0.8507 \end{bmatrix}.$$

Also, $\sigma_1^2 = (1 + \sqrt{5})/2$, and $\sigma_2^2 = (1 - \sqrt{5})/2$. It is found that $q_1^2 = i$ and $q_2^2 = -i$, and then $q_1 = \frac{\sqrt{2}}{2}(1 + i)$, $q_2 = \frac{\sqrt{2}}{2}(1 - i)$. Finally,

$$X_c = U \bar{Q} \Sigma^{1/2} = \begin{bmatrix} -0.7651(1+i) & 0.2923(1-i) \\ -0.4729(1+i) & -0.4729(1-i) \end{bmatrix}.$$

It can be verified that, indeed, $X_c X_c^T = A$.

**Appendix B. The sign characteristic and an expository example.** We first indicate that the sign characteristic is an intrinsic property of the matrix function $L(\lambda) := \lambda^2 M + \lambda D + K$, where $M$, $D$, $K$ are Hermitian and $M$ is nonsingular. For simplicity, and in the spirit of this exposition, it is assumed that all eigenvalues of $L(\lambda)$ are semisimple. The following theorem is a special case of Theorem 3.7 of [3].

Theorem 7. [illegible] $L(\lambda)$ [illegible] $\mu_1(\lambda), \ldots, \mu_n(\lambda)$ [illegible] $\lambda$ [illegible]

$$\det(\mu_j(\lambda) I - L(\lambda)) = 0, \quad j = 1, 2, \ldots, n.$$

[illegible] $\lambda_1, \ldots, \lambda_r$ [illegible] $L(\lambda)$ [illegible] $i$ $1 \le i \le r$ [illegible]

$$\mu_j(\lambda) = (\lambda - \lambda_i) \nu_{ij}(\lambda), \quad \nu_{ij}(\lambda_i) \neq 0. \tag{30}$$

[illegible] $(\nu_{ij}(\lambda_i))$ [illegible] $\lambda_i$ [illegible]

For each real eigenvalue $\lambda_i$ of $L(\lambda)$ there must be at least one function $\mu_j(\lambda)$ which vanishes at $\lambda = \lambda_i$. Thus, implicitly, (30) determines a function $\mu_j(\lambda)$ of the matrix function $L(\lambda)$ which vanishes at $\lambda_i$. Since $\mu_j'(\lambda_i) = \nu_{ij}(\lambda_i)$, the sign characteristic tells us whether this particular function has a positive or negative slope at $\lambda_i$.

Now, for illustration, consider the problem of constructing 2×2 symmetric systems with the simple (mixed) spectrum: 1, −1, i, −i. One such system is obvious, namely, the monic system

$$L(\lambda) = \begin{bmatrix} \lambda^2 - 1 & 0 \\ 0 & \lambda^2 + 1 \end{bmatrix}. \tag{31}$$

Let us first examine the [illegible] problem for this system. Using the theorem above, it is easily seen that the sign characteristic associated with the real eigenvalues $\{1,\ -1\}$ of this system is $\{+1,\ -1\}$. Then observe that this system has associated matrices

$$X = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} i & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -i \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{32}$$

Since $\det\left[\begin{smallmatrix} X \\ XJ \end{smallmatrix}\right] \neq 0$, $(X,\ J)$ form a Jordan pair. However,

$$Y := \begin{bmatrix} X \\ XJ \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ M \end{bmatrix} \neq PX^*,$$

so $(X,\ J)$ is not part of a [illegible]. The eigenvectors must be renormalized to achieve this. (Such a renormalization leaves the moments and the coefficients invariant.) It is found that if we set $\kappa = e^{-i\pi/4}$ and define

$$X = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 1 & 0 \\ \kappa & 0 & 0 & \overline{\kappa} \end{bmatrix},$$

then $Y = PX^*$. It is now a matter of computation to verify that (16) and (17) lead back to the coefficients of system (31).

Now consider the inverse problem in which the data consist of matrices $J$ and $P$ of (32). Observe first that $X_{R1}$ and $X_{R2}$ will be $2 \times 1$ vectors and, to find an $X_c$ of rank one, it is convenient to assume that $X_{R1}$ and $X_{R2}$ are linearly dependent (see section 6). Indeed, with $\alpha, \beta \in \mathbb{R}$ let us take $X_{R1} = X_{R2} = \left[\begin{smallmatrix} \alpha \\ \beta \end{smallmatrix}\right]$. Then $R_1 = 0$ in (20) and we may choose $R = \left[\begin{smallmatrix} 0 & 0 \\ 0 & \gamma \end{smallmatrix}\right]$, where $\gamma \in \mathbb{R}$, so that the equation for $X_c$ becomes $X_c X_c^T = \left[\begin{smallmatrix} 0 & 0 \\ 0 & -i\gamma \end{smallmatrix}\right]$.

It is easily verified that

$$X_c = \begin{bmatrix} 0 \\ \gamma^{\frac{1}{2}} \kappa \end{bmatrix}$$

is a solution of this equation. Thus, a complete eigenvector matrix $X$ is

$$X = \begin{bmatrix} 0 & \alpha & \alpha & 0 \\ \gamma^{\frac{1}{2}} \kappa & \beta & \beta & \overline{\gamma^{\frac{1}{2}} \kappa} \end{bmatrix}. \tag{33}$$

Now compute to find $\Gamma_0 = 0$ and

$$\Gamma_1 = 2 \begin{bmatrix} \alpha^2 & \alpha\beta \\ \alpha\beta & \beta^2 + \gamma \end{bmatrix}, \quad \Gamma_2 = 0, \quad \Gamma_3 = 2 \begin{bmatrix} \alpha^2 & \alpha\beta \\ \alpha\beta & \beta^2 - \gamma \end{bmatrix},$$

and then, assuming $\alpha\gamma \neq 0$,

$$M = \Gamma_1^{-1} = (2\alpha^2\gamma)^{-1} \begin{bmatrix} \beta^2+\gamma & -\alpha\beta \\ -\alpha\beta & \alpha^2 \end{bmatrix}, \quad D = 0,$$

$$K = (2\alpha^2\gamma)^{-1} \begin{bmatrix} \beta^2-\gamma & -\alpha\beta \\ -\alpha\beta & \alpha^2 \end{bmatrix}.$$

Thus, a three-parameter family of isospectral systems is obtained. The system (31), with which this discussion began, is obtained by taking $\alpha = 1/\sqrt{2}$, $\beta = 0$, $\gamma = 1/2$. In contrast with matrix $X$ of (32), (33) evaluated at these parameter values is a member of a [illegible] triple.

To illustrate the role played by the sign characteristic, consider the following special cases:

[illegible] 1. $\alpha = 1 \;\; \beta = 0 \;\; \gamma = 1.$

$$M_1\lambda^2 + K_1 = \frac{1}{2}\begin{bmatrix} \lambda^2-1 & 0 \\ 0 & \lambda^2+1 \end{bmatrix}.$$

[illegible] 2. $\alpha = 1 \;\; \beta = -1 \;\; \gamma = 1.$

$$M_2\lambda^2 + K_2 = \frac{1}{2}\begin{bmatrix} 2\lambda^2 & \lambda^2+1 \\ \lambda^2+1 & \lambda^2+1 \end{bmatrix}.$$

[illegible] 3. $\alpha = 1 \;\; \beta = 0 \;\; \gamma = -1.$

$$M_3\lambda^2 + K_3 = \frac{1}{2}\begin{bmatrix} \lambda^2-1 & 0 \\ 0 & -(\lambda^2+1) \end{bmatrix}.$$

Theorem 7 tells us that the sign characteristic determines the derivatives of the eigenvalue functions $\mu_1(\lambda)$, $\mu_2(\lambda)$ of the matrix function $L(\lambda)$ at the points where they cross the real axis. Obviously, in Case 1, the nature of these eigenvalue functions corresponds to the left side of Figure 1, and the eigenvalue functions are $\mu_1(\lambda) = \frac{1}{2}(\lambda^2-1)$, $\mu_2(\lambda) = \frac{1}{2}(\lambda^2+1)$. The sign characteristic $\{-1,\ +1\}$ corresponds to the sign of the derivative of $\mu_1(\lambda)$ at the points $\lambda = -1$ and $\lambda = +1$, respectively.

The system of Case 2 has a similar structure (the matrix polynomials are congruent) but now $\mu_1(\lambda) = \frac{1}{2}(\lambda^2-1)$, $\mu_2(\lambda) = \lambda^2+1$.

In contrast, Case 3 has $\mu_1(\lambda) = \frac{1}{2}(\lambda^2-1)$, $\mu_2(\lambda) = -\frac{1}{2}(\lambda^2+1)$, and $M$ is indefinite (right side of Figure 1). However, it is clear from both sides of the figure that both the real spectrum of $L(\lambda)$ and the sign characteristic are the same in every case.

**Appendix C. The rank of complex symmetric matrices.** Let $M$ be a complex symmetric matrix in $\mathbb{C}^{n\times n}$. Thus, there are real symmetric matrices $A$ and $B$ such that $X = A + iB$. Our objective is to show how the ranks of matrices $X$, $A$, and $B$ are connected. Notice that there are no other hypotheses on $A$ and $B$, such as invertibility or positivity.

The rank of a square matrix is invariant under congruence transformations so, if $S \in \mathbb{R}^{n\times n}$ is nonsingular, then $\operatorname{rank} X = \operatorname{rank}(SXS^T)$. Our problem will be resolved by applying a congruence with matrix $S$ which simultaneously reduces $A$ and $B$ to a canonical form. The canonical forms in question can be found in the recent work [11] and are described here. It is convenient to use the language of spectral analysis and consider our problem in the context of the reduction of the [illegible] $A + \lambda B$ by real congruence.

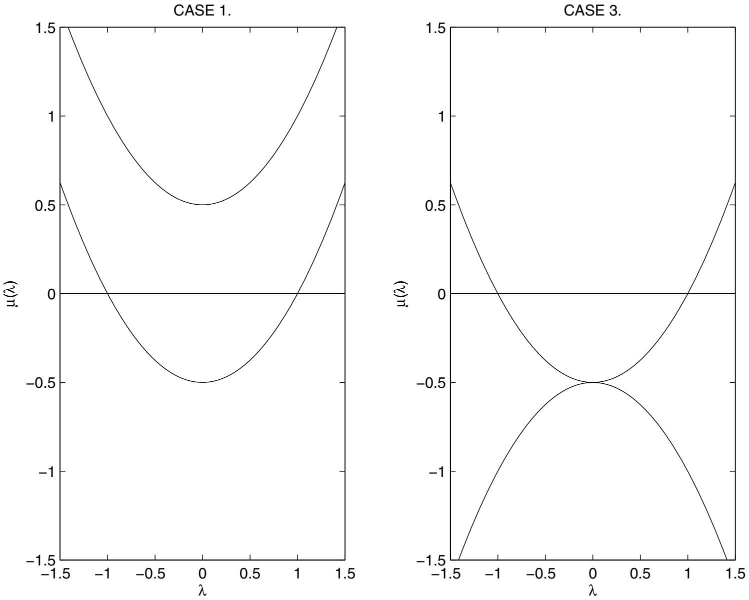


Fig. 1. *Eigenvalue functions* $\mu_j(\lambda)$.

The general canonical forms are quite complicated. They are block diagonal with blocks of several different types as follows.

- Square matrices $F_m$ of size $m$ with ones on the NE–SW diagonal and zeros elsewhere (also known as the [illegible] matrices).
- Matrices $G_m$:

$$G_m = \begin{bmatrix} 0 & \cdots & \cdots & 1 & 0 \\ \vdots & & & 0 & 0 \\ \vdots & & & & \vdots \\ 1 & 0 & & & \vdots \\ 0 & 0 & \cdots & \cdots & 0 \end{bmatrix} = \begin{bmatrix} F_{m-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

- Matrices $H_{2m}$:

$$H_{2m} = \begin{bmatrix} 0 & 0 & & \cdots & & 1 & 0 \\ 0 & & & & & 0 & -1 \\ \vdots & & & 1 & 0 & & \\ & & & 0 & -1 & & \\ & & \cdot & & & & \vdots \\ 1 & 0 & & & & & 0 \\ 0 & -1 & & & \cdots & 0 & 0 \end{bmatrix}.$$

Then the canonical form for $A$ is a direct sum of blocks of (up to) five distinct types, say,

$$A = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5,$$

and similarly for $B$. The blocks $A_r$ and $B_r$ will have the same size for each $r$ and are as follows:

1. $A_1 = B_1 = 0$ (a square zero matrix).
2. $A_2 = \sum_j \oplus G_{2\varepsilon_j+1}$,
$$B_2 = \sum_j \oplus \begin{bmatrix} 0 & 0 & F_{\varepsilon_j} \\ 0 & 0 & 0 \\ F_{\varepsilon_j} & 0 & 0 \end{bmatrix}.$$
3. $A_3 = \sum_j \oplus \delta_j F_{k_j}$, $B_3 = \sum_j \oplus \delta_j G_{k_j}$, where each $\delta_j$ is $\pm 1$ and, together, they define the sign characteristic of the eigenvalue of $A + \lambda B$ at infinity (if any).
4. $A_4 = \sum_j \oplus(\eta_j \alpha_j F_{l_j} + G_{l_j})$, $B_4 = \sum_j \oplus \eta_j F_{l_j}$, where each $\eta_j$ is $\pm 1$ and, together, they define the sign characteristic associated with the real eigenvalues (if any).
5. $$A_5 = \sum_j \oplus \left( \mu_j F_{2m_j} + \nu_j H_{2m_j} + \begin{bmatrix} F_{2m_j - 2} & 0 \\ 0 & 0_2 \end{bmatrix} \right), \qquad B_5 = \sum_j \oplus F_{2m_j},$$
where $\mu_j \pm i\nu_j$ with $\nu_j \neq 0$ are the complex eigenvalues (if any), and $0_2$ is the $2 \times 2$ zero matrix.

It is clear that $\operatorname{rank}(A) = \sum_{r=1}^{5} \operatorname{rank}(A_r)$, $\operatorname{rank}(B) = \sum_{r=1}^{5} \operatorname{rank}(B_r)$, and $\operatorname{rank}(A + iB) = \sum_{r=1}^{5} \operatorname{rank}(A_r + iB_r)$. However, the ranks of these component matrices are easily obtained from those of the $F$'s, $G$'s, and $H$'s. Notice, in particular, that for the first four types the component diagonal blocks are triangular, and the rank can be read off by observation. For the fifth type, the structures will be clear if we just examine the case $m_j = 3$ more closely. It is easily seen that, if we define

$$\Delta_j := \begin{bmatrix} \nu_j & \mu_j \\ \mu_j & -\nu_j \end{bmatrix},$$

then, in this case,

$$A_5 = \begin{bmatrix} 0_2 & F_2 & \Delta_j + iF_2 \\ F_2 & \Delta_j + iF_2 & 0_2 \\ \Delta_j + iF_2 & 0_2 & 0_2 \end{bmatrix},$$

so that $\det(A_5) = 0$ if and only if $\det(\Delta_j + iF_2) = 0$ (and this is the case whatever the value of $m_j$). However,

$$\det(\Delta_j + iF_2) = -(\mu_j^2 + \nu_j^2) - 2i\mu_j + 1,$$

and this vanishes if and only if $\mu_j = 0$ and $\nu_j = \pm 1$; i.e., the corresponding complex eigenvalue pair is $\pm i$.

Notice also that, when $\mu_j + i\nu_j = i$, then

$$\Delta_j + iF_2 = \begin{bmatrix} 1 & i \\ i & -1 \end{bmatrix},$$

a matrix of rank one.

The important conclusion of this argument is the following.

PROPOSITION 8.

$$\operatorname{rank}(A_5) = 2\sum_j m_j, \tag{34}$$

[illegible] $j$ [illegible] $\pm i$

[illegible] $A_5$ [illegible] (34) [illegible] $i$ [illegible] $-i$

It will be convenient to denote the algebraic multiplicity of the eigenvalue $i$ by $a(i)$ so that, in general,

$$\operatorname{rank}(A_5) = 2\sum_j m_j - a(i).$$

Now it can be seen that

1. $\operatorname{rank}(A_1) = \operatorname{rank}(B_1) = \operatorname{rank}(A_1 + iB_1) = 0$;
2. $\operatorname{rank}(A_2) = \operatorname{rank}(B_2) = \operatorname{rank}(A_2 + iB_2) = \sum_j 2\varepsilon_j$;
3. $\operatorname{rank}(A_3) = \sum_j k_j$, $\operatorname{rank} B_3 = \sum_j (k_j - 1)$, $\operatorname{rank}(A_3 + iB_3) = \sum_j k_j$;
4. $\operatorname{rank}(A_4) = \sum_{j:\alpha_j \neq 0} l_j + \sum_{j:\alpha_j = 0}(l_j - 1)$, $\operatorname{rank}(B_4) = \sum_j l_j$, $\operatorname{rank}(A_4 + iB_4) = \sum_j l_j$;
5. $\operatorname{rank}(A_5) = 2\sum_j m_j - a(i)$, $\operatorname{rank}(B_5) = 2\sum_j m_j$, $\operatorname{rank}(A_5 + iB_5) = 2\sum_j m_j$.

Now consider how the ranks of $A$ and $B$ can differ from that of $A + iB$. Items 1 and 2 produce no differences. Due to item 3, however, the rank of $B_3$ (and hence $B$) is less than the other two ranks by one for [illegible] associated with the eigenvalue at infinity (if there is such an eigenvalue), i.e., the geometric multiplicity of the eigenvalue at infinity, say $g(\infty)$.

Similarly, it can be deduced from item 4 that $\operatorname{rank}(A_4)$ is less than $\operatorname{rank}(B_4)$ and $\operatorname{rank}(A_4 + iB_4)$ by the geometric multiplicity of the zero eigenvalue, say, $g(0)$. The case of blocks of the fifth type is covered by Proposition 8.

Since $\operatorname{rank}(A) = \sum_{r=1}^{5} \operatorname{rank}(A_r)$ and $\operatorname{rank}(B) = \sum_{r=1}^{5} \operatorname{rank}(B_r)$, the results can be brought together in the following form (and we keep in mind that $a(i)$, $g(o)$, and $g(\infty)$ refer to eigenvalues of the pencil $A + \lambda B$).

THEOREM 9.

$$\operatorname{rank}(A + iB) = \operatorname{rank}(A) - a(i) + g(0) = \operatorname{rank}(B) - a(i) + g(\infty).$$

[illegible] Let us illustrate with Example 4 of the main text. The first case arising there is

$$A + iB = \frac{1}{2}\begin{bmatrix} -1 & -i \\ -i & 1 \end{bmatrix},$$

so that

$$A + \lambda B = \frac{1}{2}\begin{bmatrix} -1 & -\lambda \\ -\lambda & 1 \end{bmatrix}.$$

We have $a(i) = 1$, $g(0) = g(\infty) = 0$, and $\operatorname{rank}(A + iB) = 1$.

The second case is

$$A + iB = \begin{bmatrix} 0 & 0 \\ 0 & i \end{bmatrix},$$

so that

$$A + \lambda B = \begin{bmatrix} 0 & 0 \\ 0 & \lambda \end{bmatrix}.$$

Notice first that, in this case, $A_1 = B_1 = [0]$. Then $a(i) = 0$, $g(0) = 1$, $g(\infty) = 0$, and $\operatorname{rank}(A + iB) = 1$. ◻

**Acknowledgment.** The author is grateful to Prof. N. J. Higham for the provision of an excellent research environment.

## REFERENCES

[1] A. Bunse-Gerstner and W. B. Gragg, *Singular value decompositions of complex symmetric matrices*, J. Comput. Appl. Math., 21 (1988), pp. 41–54.

[2] R. J. Duffin, *A minimax theory for overdamped networks*, J. Rational Mech. Anal., 4 (1955), pp. 221–233.

[3] I. Gohberg, P. Lancaster, and L. Rodman, *Spectral analysis of selfadjoint matrix polynomials*, Ann. of Math. (2), 112 (1980), pp. 33–71.

[4] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[5] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[6] C.-H. Guo and P. Lancaster, *Algorithms for hyperbolic quadratic eigenvalue problems*, Math. Comp., 74 (2005), pp. 1777–1791.

[7] N. J. Higham, F. Tisseur, and P. M. Van Dooren, *Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems*, Linear Algebra Appl., 351/352 (2002), pp. 455–474.

[8] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[9] P. Lancaster, *Isospectral vibrating systems. Part* 1: *The spectral method*, Linear Algebra Appl., 409 (2005), pp. 51–69.

[10] P. Lancaster and U. Prells, *Inverse problems for damped vibrating systems*, J. Sound Vibration, 283 (2005), pp. 891–914.

[11] P. Lancaster and L. Rodman, *Canonical forms for Hermitian matrix pairs under strict equivalence and congruence*, SIAM Rev., 47 (2005), pp. 407–443.

[12] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, Orlando, 1985.

[13] P. Lancaster and M. Tismenetsky, *Inertia characteristics of self-adjoint matrix polynomials*, Linear Algebra Appl., 52/53 (1983), pp. 479–496.

[14] P. Lancaster and Q. Ye, *Inverse spectral problems for linear and quadratic matrix pencils*, Linear Algebra Appl., 107 (1988), pp. 293–309.

[15] U. Prells and P. Lancaster, *Isospectral vibrating systems.* II. *Structure preserving transformations*, in Operator Theory and Indefinite Inner Product Spaces, Oper. Theory Adv. Appl. 163, Birkhäuser, Basel, 2006, pp. 275–298.

[16] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 302–327

# UNSYMMETRIC ORDERING USING A CONSTRAINED MARKOWITZ SCHEME*

PATRICK R. AMESTOY†, XIAOYE S. LI‡, AND STÉPHANE PRALET†

**Abstract.** We present a family of ordering algorithms that can be used as a preprocessing step prior to performing sparse **LU** factorization. The ordering algorithms simultaneously achieve the objectives of selecting numerically good pivots and preserving the sparsity. We describe the algorithmic properties and challenges in their implementation. By mixing the two objectives we show that we can reduce the amount of fill-in in the factors and reduce the number of numerical problems during factorization. On a set of large unsymmetric real problems, we obtained the median reductions of 12% in the factorization time, of 13% in the size of the **LU** factors, of 20% in the number of operations performed during the factorization phase, and of 11% in the memory needed by the multifrontal solver `MA41_UNS`. A byproduct of this ordering strategy is an incomplete **LU**-factored matrix that can be used as a preconditioner in an iterative solver.

**Key words.** sparse unsymmetric matrices, greedy heuristics, ordering methods, bipartite quotient graph

**AMS subject classifications.** 05C50, 65F05, 65F50

**DOI.** 10.1137/050622547

## 1. Introduction.

Direct methods for sparse unsymmetric linear systems usually involve an analysis phase preceding the effective **LU** factorization [2, 10, 14, 20, 21]. The analysis phase transforms $\mathbf{A}$ into $\tilde{\mathbf{A}}$ with better properties for sparse factorization. It exploits the structural information to reduce the amount of fill-in in the **LU** factors and exploits the numerical information to reduce the need for numerical pivoting during factorization.

Two separate steps can be used in sequence for these two objectives:

1. Scaling and maximum transversal algorithms are used to transform $\mathbf{A}$ into $\mathbf{A}_1$ with large entries in magnitude on the diagonal.
2. A symmetric fill-reducing ordering, which preserves the large diagonal, is used to permute $\mathbf{A}_1$ into $\tilde{\mathbf{A}}$ so that the factors of $\tilde{\mathbf{A}}$ are sparser than those of $\mathbf{A}_1$.

Thus, the ultimate factorization is

$$\mathbf{LU} = \mathbf{P}_3\mathbf{P}_2\mathbf{D}_r\mathbf{A}\mathbf{D}_c\mathbf{Q}_1\mathbf{P}_2^T\mathbf{Q}_3, \tag{1.1}$$

where $\mathbf{D}_r$ and $\mathbf{D}_c$ are diagonal scaling matrices, $\mathbf{Q}_1$ is a permutation obtained from the maximum transversal algorithm, $\mathbf{P}_2$ corresponds to the fill-reducing permutation, and $\mathbf{P}_3$ and $\mathbf{Q}_3$ are permutations corresponding to numerical pivoting during factorization.

It has been observed in [4] that permuting large entries on the diagonal (computing $\mathbf{Q}_1$ based on [17]) can significantly reduce the number of numerical problems

---

*Received by the editors August 17, 2005; accepted for publication (in revised form) August 10, 2006; published electronically February 13, 2007. Part of this research was supported by NSF-INRIA grant NSF-INT-0003274.

http://www.siam.org/journals/simax/29-1/62254.html

†ENSEEIHT-IRIT, 2, rue Camichel, BP 7122 - F 31071 Toulouse Cedex 7, France (Patrick.Amestoy@enseeiht.fr, Stephane.Pralet@enseeiht.fr). The work of the third author was partially supported by CERFACS, 42, av. G. Coriolis, 31057 Toulouse Cedex 01, France, during his Ph.D.

‡Lawrence Berkeley National Lab, MS 50F-1650, 1 Cyclotron Rd., Berkeley, CA 94720 (xsli@lbl.gov). The work of the second author was supported in part by the Director, Office of Advanced Scientific Computing Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract DE-AC03-76SF00098, and was supported in part by the National Science Foundation Cooperative Agreement ACI-9619020, NSF grant ACI-9813362.

during factorization. A standard way to find $\mathbf{P}_2$ is to apply a symmetric ordering algorithm (e.g., `AMD` [1]) to the structure of $\mathbf{A}_1 + \mathbf{A}_1^T$, where $\mathbf{A}_1 = \mathbf{D}_r\mathbf{A}\mathbf{D}_c\mathbf{Q}_1$. A better algorithm, called [illegible] (or `DMLS`), was developed in [5], which could exploit the unsymmetric structure of $\mathbf{A}_1$ and was shown to give sparser factors than with `AMD`.

The above two-step approach has two drawbacks:

- The numerical treatment forces the fill-reducing ordering to restrict pivot selection on the diagonal of $\mathbf{A}_1$, and so to compute a symmetric permutation.
- The ordering phase does not have numerical information to select pivots.

To improve sparsity preservation and numerical quality of the preselected pivots, we describe in this paper a family of orderings that can select off-diagonal pivots using a combination of structural and numerical criteria. Based on a numerical preprocessing of the matrix we build a set of numerically acceptable pivots, referred to as matrix $\mathbf{C}$, that may contain off-diagonal entries. We then compute an unsymmetric ordering taking into account both the structure of $\mathbf{A}$ and the numerical information in $\mathbf{C}$. The $\mathbf{C}$ matrix serves as a [illegible] for the pivot selection, and nontrivial floating point operations can be performed on this matrix to update the characteristics of the pivots. The new algorithm is referred to as [illegible] (or `CMLS`).

In summary, this work extended and generalized the `DMLS` work in several ways:

1. We do not limit our choice of pivots to the maximum transversal of $\mathbf{D}_r\mathbf{A}\mathbf{D}_c$. Our pivots can be chosen from a constraint matrix $\mathbf{C}$ that includes a transversal but is not limited to this transversal.
2. The constraint matrix $\mathbf{C}$ is updated [illegible] after each step of elimination. The final $\mathbf{C}$ is an incomplete $\mathbf{LU}$ factor of $\mathbf{D}_r\mathbf{A}\mathbf{D}_c$.

Thus, instead of computing the permutations $\mathbf{Q}_1$ and $\mathbf{P}_2$ of (1.1) in two separate steps, `CMLS` simultaneously computes row and column permutations $\mathbf{P}_2$ and $\mathbf{Q}_2$, and the final factorization is

$$\mathbf{LU} = \mathbf{P}_3\mathbf{P}_2\mathbf{D}_r\mathbf{A}\mathbf{D}_c\mathbf{Q}_2^T\mathbf{Q}_3. \tag{1.2}$$

We evaluated the new ordering algorithms using two state-of-the-art direct solvers: the multifrontal code `MA41_UNS` [2, 7] and the supernodal code `SuperLU_DIST` [27, 28]. In `MA41_UNS`, standard partial pivoting with a threshold value is applied to locally select numerically stable pivots within a so-called frontal matrix. It is possible that some variables cannot be locally eliminated and are postponed for later eliminations, which may result in an increase in the size of the $\mathbf{LU}$ factors and the number of operations compared with those predicted during analysis. In `SuperLU_DIST`, a static pivoting strategy is used and the pivotal sequence chosen during analysis is kept the same (i.e., $\mathbf{P}_3$ and $\mathbf{Q}_3$ are the identity matrices in (1.1) and (1.2)). Iterative refinement may be needed to improve the solution.

The rest of the paper is organized as follows. Section 2 introduces the main components of our algorithm. Section 3 defines the graph-theoretic notation and describes the use of local symmetrization in our context. Section 4 describes the algorithmic contributions of the proposed `CMLS` method. A full detailed presentation of our implementation is given in [32]. Section 5 analyzes the results of the newly implemented `CMLS` algorithm when applied to real-life unsymmetric test cases.

**2. Components of our unsymmetric ordering.** Given a matrix $\mathbf{A}$, let $\mathcal{P}attern(\mathbf{A})$ be the set of nonzero entries of $\mathbf{A}$: $\mathcal{P}attern(\mathbf{A}) = \{(i, j) \text{ such that } a_{ij} \neq 0\}$. Our unsymmetric ordering consists of two main steps:

- *Step* 1. Based on a numerical pretreatment of the matrix $\mathbf{A}$, we extract a set of numerically acceptable pivots, referred to as the *constraint matrix* $\mathbf{C}$. We have $\mathcal{P}attern(\mathbf{C}) \subseteq \mathcal{P}attern(\mathbf{A})$, and if $c_{ij} \neq 0$ then $c_{ij} = a_{ij}$.
- *Step* 2. Constrained unsymmetric ordering: the constraint matrix is used at each step of the symbolic Gaussian elimination to control the set of eligible pivots (possibly with respect to both numerical and structural criteria).

Before describing these two steps more precisely, we introduce definitions and notation that will be used to describe our algorithms.

Let $\mathbf{M} = (m_{ij})$ be a matrix of order $n$. If $\mathbf{M}$ can be permuted to have $n$ nonzeros on the diagonal then $\mathbf{M}$ is *structurally nonsingular*. Let $G_M = (V_r, V_c, E)$ be the bipartite graph associated with the matrix $\mathbf{M}$. $V_r$ is the set of row vertices and $V_c$ is the set of column vertices. Let $(i, j) \in V_r \times V_c$; then $(i, j) \in E$ if and only if $m_{ij} \neq 0$. A *matching* is a subset of edges $\mathcal{M} \subseteq E$ such that for all vertices $v \in V_r \cup V_c$, at most one edge of $\mathcal{M}$ is incident on $v$. If $\mathbf{M}$ is structurally nonsingular, then there exists a matching $\mathcal{M}$ with $n$ edges and $\mathcal{M}$ is said to be a *perfect matching*. We will also say that $\mathcal{M}$ is a *perfect transversal*.

For the sake of clarity, in the remainder of this paper we assume that $\mathbf{A}$ is structurally nonsingular. The adaptation of our algorithms to structurally singular matrices is straightforward but would have severely complicated our notation and comments.

### 2.1. Step 1: Numerical preprocessing.

The objective of this preprocessing step is to extract the most significant (structurally and numerically) entries of the matrix $\mathbf{A}$ and to use them to build the constraint matrix $\mathbf{C}$.

First, we scale the matrix $\mathbf{A}$ with the diagonal matrices $\mathbf{D}_r$ and $\mathbf{D}_c$, resulting in $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$. The objective of this scaling is to homogenize the magnitude of the entries of the matrix. In particular, it helps us to compare the magnitude of entries belonging to different rows and columns and thus to decide which entries will be selected in our *constraint matrix* $\mathbf{C}$. In this paper we use the scaling computed with the maximum weighted matching algorithm [17]. All entries in our scaled matrix then have entries lower than 1 in magnitude with a perfect transversal with entries of magnitude equal to 1.

Second, a *constraint matrix* $\mathbf{C}$ can be constructed from $\mathbf{A}$ such that $\mathcal{P}attern(\mathbf{C}) \subseteq \mathcal{P}attern(\mathbf{A})$ and $\mathbf{C}$ satisfies certain numerical and/or structural properties. Since the entries in $\mathbf{C}$ correspond to the potential pivots for the subsequent step, we keep only a subset of bounded size (typically less than $3n$) of the largest entries in the scaled matrix. Furthermore, we want $\mathbf{C}$ to be structurally nonsingular, and thus we add entries from $\mathbf{A}$ to guarantee that $\mathbf{C}$ includes a perfect transversal $\mathcal{M}$.

### 2.2. Step 2: Constrained unsymmetric ordering.

Let $\mathbf{A}^1 = \mathbf{A}$ be the original matrix of order $n$ and $\mathbf{A}^k$ be the reduced matrix after eliminating the first $k-1$ pivots (not necessarily on the diagonal). Let $\mathbf{C}^1 = \mathbf{C}$ be such that $\mathcal{P}attern(\mathbf{C}^1) \subseteq \mathcal{P}attern(\mathbf{A}^1)$. At each step $k$, a pivot $p^k$ such that $p^k \in \mathcal{P}attern(\mathbf{C}^k)$ is selected. This selection may combine structural heuristics based on the structure of $\mathbf{A}^k$ (e.g., approximate Markowitz count, approximate minimum fill, etc.) and numerical heuristics carried by the $\mathbf{C}^k$ matrix. Matrix $\mathbf{A}^k$ is updated (remove the row and the column of the pivot and add fill-ins in the Schur complement). Matrix $\mathbf{C}^k$ is updated such that $\mathbf{C}^{k+1}$ remains structurally nonsingular and $\mathcal{P}attern(\mathbf{C}^{k+1})$ is included in $\mathcal{P}attern(\bar{\mathbf{C}}^k)$, where $\bar{\mathbf{C}}^k$ is defined as the reduced matrix after the elimination of pivot $p^k$ in $\mathbf{C}^k$. The structure of $\mathbf{A}^{k+1}$ contains the structure of $\bar{\mathbf{C}}^k$. This implies that $\mathcal{P}attern(\mathbf{C}^{k+1}) \subseteq \mathcal{P}attern(\mathbf{A}^{k+1})$. To keep $\mathbf{C}^k$ structurally nonsingular, a perfect matching in $\mathbf{C}^k$ is maintained at each step. When there is no ambiguity, we will omit the superscript $k$ from the matrix notation.

The following two considerations influence the update that will be performed on $\mathbf{C}$:

- Which metric do we use to select a pivot?
- Which entries and/or values are added/updated in $\mathbf{C}$ at each step of the elimination?

Note that if we consider the magnitude of $\mathbf{C}$'s entries to select a pivot, both the pattern of $\mathbf{C}$ and the numerical values need to be stored and updated. Furthermore, any structural information about each entry $(i, j)$ in $\mathbf{C}$ should carry information on the reduced matrix associated with the complete matrix $\mathbf{A}$.

The ordering algorithm also depends on how $\mathbf{C}$ is updated at each step. As mentioned before in the description of Step 2, we want at each step to guarantee that

$$\mathbf{C} \text{ must remain structurally nonsingular, and} \tag{2.1}$$

$$\mathcal{P}attern(\mathbf{C}^{k+1}) \subseteq \mathcal{P}attern(\bar{\mathbf{C}}^k). \tag{2.2}$$

## 3. Notation and definitions.

Before giving the algorithmic details of the proposed `CMLS` method, we introduce the graph structures and notation that will be used in this paper. We first describe the main properties of bipartite graphs and bipartite quotient graphs and their relationship with Gaussian elimination. We then introduce the notation that will be used to describe our algorithms and define local symmetrization [5], a technique that simplifies the bipartite quotient graph implementation. Note that we use calligraphic font for notation related to quotient graphs and Roman font for other graphs.

### 3.1. Bipartite graph.

Let $\mathbf{M} = (m_{ij})$ be a matrix and $G_M = (V_r, V_c, E)$ be its associated bipartite graph. Let $R_i$ denote the structure of row $i$; i.e., $R_i = \{j \in V_c \text{ s.t. } (i, j) \in E\}$. Let $C_j$ denote the structure of column $j$; i.e., $C_j = \{i \in V_r \text{ s.t. } (i, j) \in E\}$.

In Gaussian elimination, when a pivot $p = (r_p, c_p)$ is eliminated, a new matrix, referred to as the reduced matrix $\bar{\mathbf{M}}$, is computed. $\bar{\mathbf{M}}$ is obtained from $\mathbf{M}$ by removing row $r_p$ and column $c_p$ and by adding the Schur complement entries. In terms of graph manipulations, this elimination adds edges in the bipartite graph of $\mathbf{M}$ to connect all the rows adjacent to $c_p$ to all the columns adjacent to $r_p$. This set of connected rows and columns is referred to as a *biclique*.

The symbolic factorization of $\mathbf{M}$ is done by building $\mathbf{M}^k$ for $k = 1$ to $n$, with $\mathbf{M}^1 = \mathbf{M}$. After eliminating the $k$th pivot, we compute $\mathbf{M}^{k+1} = \bar{\mathbf{M}}^k$.

### 3.2. Bipartite quotient graph.

In the previous section we have shown that to update the bipartite graph we must add, at each elimination step, entries to the Schur complement matrix which may be costly to update and to store. It has been shown that quotient graphs can be used to efficiently model the factorization of symmetric matrices [20, 24]. The main idea is to use a compact representation of the cliques associated with the eliminated vertices. This concept can be extended (see [31]) to model the $\mathbf{LU}$ factorization. In this case, a bipartite quotient graph can be used to represent the edges in a biclique. It has then been shown in [31] that doing so the elimination can be modeled in space bounded by the size of the original matrix $\mathbf{A}$.

In this section, we first explain why the quotient graph model leads to more complex algorithms on unsymmetric matrices than on symmetric matrices. We then briefly define element absorption and explain the use of local symmetrization to reduce the quotient graph complexity. Finally, we introduce notation that will be used to describe our algorithms.

Let $\mathcal{P}_r \subseteq 2^{V_r}$ and $\mathcal{P}_c \subseteq 2^{V_c}$ be two partitions of $V_r$ and $V_c$, respectively. We define the bipartite quotient graph $\mathcal{G}_A = (\mathcal{P}_r, \mathcal{P}_c, \xi)$ of $\mathbf{A}$ such that an edge $(\mathcal{I}, \mathcal{J})$ belongs to $\xi \subseteq \mathcal{P}_r \times \mathcal{P}_c$ if and only if there exists an edge in $G$ between a node of $\mathcal{I}$ and a node of $\mathcal{J}$.

Let $\mathcal{G}_A^k$ be the bipartite quotient graph used to represent the structure of the reduced submatrix $\mathbf{A}^k$ after $k$ steps of elimination. Initially the bipartite quotient graph $\mathcal{G}_A^1$ is initialized with the partitions $\mathcal{P}_r = \{\{i\}$ such that $i \in V_r\}$ and $\mathcal{P}_c = \{\{j\}$ such that $j \in V_c\}$. Thus it is equivalent to the bipartite graph $G^1$. At step $k$ of Gaussian elimination, any eliminated pivot $e = (r_e, c_e)$ will be referred to as a [illegible]. All the row and column vertices that are not coupled elements are referred to as the row and column [illegible] of $\mathcal{G}_A^k$. Both row and column vertices of the graph are thus partitioned into two sets composed of variables (uneliminated vertices) and [illegible] (eliminated vertices). We then define $\mathcal{G}_A^k = (\mathcal{V}_r^k \cup \overline{\mathcal{V}}_r^k, \mathcal{V}_c^k \cup \overline{\mathcal{V}}_c^k, \mathcal{E}^k \cup \overline{\mathcal{E}}^k)$. When it is clear from the context, we will omit the superscript $k$. The vertices in $\mathcal{V}_r$ (resp., $\mathcal{V}_c$) correspond to the row (resp., column) variables. The vertices in $\overline{\mathcal{V}}_r$ (resp., $\overline{\mathcal{V}}_c$) correspond to the row (resp., column) elements. The edge set $\mathcal{E}$ is such that $\mathcal{E} \subseteq (\mathcal{V}_r \times \mathcal{V}_c)$, whereas $\overline{\mathcal{E}}$ is such that $\overline{\mathcal{E}} \subseteq (\mathcal{V}_r \times \overline{\mathcal{V}}_c) \cup (\overline{\mathcal{V}}_r \times \mathcal{V}_c) \cup (\overline{\mathcal{V}}_r \times \overline{\mathcal{V}}_c)$. With our definitions $(i, j)$ is a nonzero entry in the reduced matrix at step $k$ if and only if there exists a path joining $i$ and $j$ which visits only the elements and for which all the edges in the even positions correspond to already eliminated pivots. In other words, the structure of a row $i$ at step $k$ is the set of reachable columns $j$ through all the paths of the form $i \to c_{e_1} \to r_{e_1} \cdots \to c_{e_l} \to r_{e_l} \to j$, where $e_t = (r_{e_t}, c_{e_t}), 1 \leq t \leq l$, are coupled elements. Similarly, the structure of a column $j$ at step $k$ is the set of reachable rows $i$ through all the paths of the form $j \to r_{e_1} \to c_{e_1} \cdots \to r_{e_l} \to c_{e_l} \to i$. This process may involve paths of arbitrary length in $\mathcal{G}_A^k$ [31] and in particular through more than one coupled element. For example, in Figure 3.1, we assume that the entry $(r_p, c_{e_2})$ is initially zero and corresponds to fill-in due to the elimination of element $e_1$. Because of the path $r_p \to c_{e_1} \to r_{e_1}$, we know that the row structure of $r_p$ contains the row structure of $e_1$ and in particular the entry $(r_p, c_{e_2})$. We know also that the row structure of $r_p$ contains the row structure of $e_2$ because of the path $r_p \to c_{e_1} \to r_{e_1} \to c_{e_2} \to r_{e_2}$.

In the context of sparse Cholesky factorization, an undirected quotient graph (the row and column vertices are merged) is preferred and commonly used to compute an ordering for symmetric matrices (e.g., multiple minimum degree [29] and approximate minimum degree [1]). The structure of the factors can be computed following the paths of length at most two in this quotient graph. There are no edges between the elements.

In the unsymmetric case, when a pivot $p = (r_p, c_p)$ is selected, if there exists a cycle of the form $r_p \to c_{e_1} \to r_{e_1} \cdots \to c_{e_l} \to r_{e_l} \to c_p \to r_p$, then, except for $r_p$ and $c_p$, the row and column elements in the cycle are no longer needed to retrieve the structure of the remaining variables. This process will be called [illegible] and is illustrated in Figure 3.1. This absorption can be explained by the two following remarks (see [31] for further details):

- The row and the column of $p$ contain the structures of, respectively, the row

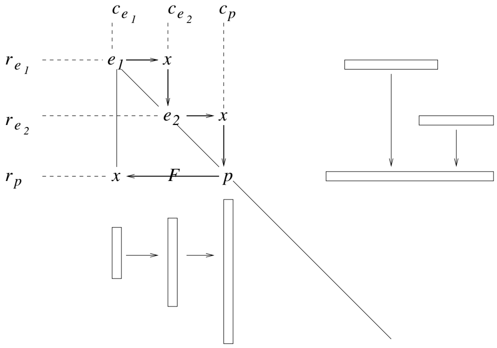


FIG. 3.1. *Illustration of a cycle* ($r_p \rightarrow c_{e_1} \rightarrow r_{e_1} \rightarrow c_{e_2} \rightarrow r_{e_2} \rightarrow c_p \rightarrow r_p$ ).

elements and the column elements in the cycle.

- If one of the elements is reachable from a variable $i$, then the other elements in the cycle are also reachable from $i$ (in particular $p$).

During the absorption, each path from $i$ to an element in the cycle is thus replaced by an edge from $i$ to the current pivot.

To avoid long search paths when we compute the structure of the row and the column of a pivot we decided to relax the element absorption rule as done in [5]. A row (resp., column) element is absorbed by the current row (resp., column) pivot if either it is adjacent to the column (resp., row) pivot or its associated column (resp., row) element is adjacent to the row (resp., column) pivot. This is referred to as [illegible] in [5]. It implies that the resulting quotient graph $\mathcal{G}_A^k$ at step $k$ models only an approximation of the structure of the reduced submatrix. It has been shown in [5] that the exploitation of element absorption combined with local symmetrization results in an in-place algorithm: at each step of the Gaussian elimination, the size of the quotient graph is bounded by the size of $\mathcal{G}_A^1$. Note that because of local symmetrization, an approximation of the symbolic factors can be computed following the paths of length at most three of the form $i \rightarrow c_e \rightarrow r_e \rightarrow j$, where $(r_e, c_e)$ denotes a coupled row and column element. Note that applying local symmetrization is significantly different from symmetrization of the complete matrix. In fact we add at most $n-1$ virtual entries (at most one per absorbed element), and thus the structure of the factors computed with local symmetrization is equal to the real structure of the factors of a matrix $\mathbf{A}+\mathbf{D}$ where $\mathbf{D}$ has fewer than $n$ entries.

To simplify the description of how the bipartite quotient graph is modified at each elimination step, we define $\overline{\mathcal{V}} \subseteq (\overline{\mathcal{V}}_r \times \overline{\mathcal{V}}_c)$ to be the set of coupled row and column elements corresponding to already eliminated pivots. Entries of the set $\overline{\mathcal{V}}$ will also be referred to as [illegible] or [illegible] when it is clear from the context. Let $\mathcal{U}_p$ (resp., $\mathcal{L}_p$) be the column (resp., row) variables adjacent in $\mathcal{G}_A^k$ to the row (resp., column) element of a pivot $p = (r_p, c_p)$. Thanks to local symmetrization, the concept of absorption can be extended to coupled elements: an element $e = (r_e, c_e)$ such that $(r_p, c_e) \in \overline{\mathcal{E}}$ [illegible] $(r_e, c_p) \in \overline{\mathcal{E}}$ can be absorbed by $p$ when $p$ is selected as a pivot. A consequence of this absorption is that our ordering also generates a dependency graph between elements that is in fact a forest. This forest will be fully exploited by the unsymmetrized multifrontal approach [7].

For each row variable $i \in \mathcal{V}_r$ and column variable $j \in \mathcal{V}_c$, we define the element lists $\mathcal{R}_i$ and $\mathcal{C}_j$ as follows:

$$\mathcal{R}_i = \{e = (r_e, c_e) \in \overline{\mathcal{V}} \text{ s.t. } (i, c_e) \in \overline{\mathcal{E}}\}$$

and

$$\mathcal{C}_j = \{e = (r_e, c_e) \in \overline{\mathcal{V}} \text{ s.t. } (r_e, j) \in \overline{\mathcal{E}}\}.$$

Let $e = (r_e, c_e)$ be an element; if $e \in \mathcal{R}_i$ then we will say that [illegible] $e$ [illegible] $i$. Similarly, if $e \in \mathcal{C}_j$ we will say that [illegible] $e$ [illegible] $j$.

Using this notation, the adjacency of a row variable $i$ (resp., column $j$) in $\mathcal{G}_A$ consists of a list of column variables denoted as $\mathcal{A}_{i*}$ (resp., a list of row variables $\mathcal{A}_{*j}$) and a list of elements $\mathcal{R}_i$ (resp., $\mathcal{C}_j$). Initially $\mathcal{R}_i = \mathcal{C}_j = \emptyset$ and $\mathcal{A}_{i*}$ and $\mathcal{A}_{*j}$ correspond to the original entries of $\mathbf{A}$. Each step of Gaussian elimination involves changes in the sets $\mathcal{R}_i$ and $\mathcal{C}_j$ as well as the computation of the structure of a current pivot $p$. The variable lists $\mathcal{A}_{i*}$ and $\mathcal{A}_{*j}$ can also be pruned. Indeed, the edges in $\mathcal{G}_A$ between the variables and the elements implicitly represent the biclique of the element and can thus be used to remove the redundant entries in $\mathcal{A}_{i*}$ and $\mathcal{A}_{*j}$. This important point will be further discussed in detail in section 4.3.

When $(r_p, c_p) \in \mathcal{V}_r \times \mathcal{V}_c$ is selected as the next pivot we build the element $p$ such that

$$\mathcal{U}_p = \mathcal{A}_{r_p*} \cup \bigcup_{e \in \mathcal{R}_{r_p}} \mathcal{U}_e \cup \bigcup_{e \in \mathcal{C}_{c_p}} \mathcal{U}_e \tag{3.1}$$

and

$$\mathcal{L}_p = \mathcal{A}_{*r_p} \cup \bigcup_{e \in \mathcal{C}_{c_p}} \mathcal{L}_e \cup \bigcup_{e \in \mathcal{R}_{r_p}} \mathcal{L}_e. \tag{3.2}$$

The third term in each equation results from local symmetrization and will enable the current pivot to absorb all the elements which it was adjacent to. For example, let us assume that the entry $p1$ is selected as pivot in Figure 3.2. Since $c_{p1}$ is adjacent to $e_1$, local symmetrization adds the virtual $S_{p1}$ entry so that the row structure of $p1$ contains $\mathcal{U}_{e_1}$.

Let $\mathcal{F}_p = \mathcal{C}_{c_p} \cup \mathcal{R}_{r_p}$ be the set of elements adjacent to the current pivot. The elements in $\mathcal{F}_p$ are absorbed by $p$ and the adjacency of each column variable $j$ in $\mathcal{U}_p$ (resp., $i$ in $\mathcal{L}_p$) is updated so that $\mathcal{C}_j \leftarrow (\mathcal{C}_j \setminus \mathcal{F}_p) \cup \{p\}$ (resp., $\mathcal{R}_i \leftarrow (\mathcal{R}_i \setminus \mathcal{F}_p) \cup \{p\}$). The structure of column $j$ of the factors in the reduced matrix is then given by $\mathcal{A}_{*j} \cup \bigcup_{e \in \mathcal{C}_j} \mathcal{L}_e$. The structure of row $i$ of the factors is $\mathcal{A}_{i*} \cup \bigcup_{e \in \mathcal{R}_i} \mathcal{U}_e$.

Note that, although the above structural changes of the reduced submatrix are correct, they should not be used to estimate the structure of the factors. Indeed, if $(i, j)$ were selected as the next pivot, then the correctly computed structure of the reduced matrix should include the local symmetrization terms (similar to (3.1) and (3.2)). In Figure 3.2, we illustrate the effect of local symmetrization on the structure of the selected pivot. Let us consider two candidate pivots belonging to the same row $r_p$, $p1 = (r_p, c_{p1})$ and $p2 = (r_p, c_{p2})$. We assume that all the elements in $\mathcal{G}_A$ adjacent to $p1$ and $p2$ are indicated in the figure. The structure of row $r_p$ is then given by $\mathcal{A}_{r_p*} \cup \mathcal{U}_{e_3}$. This, however, does not give enough information on the structure of row $r_p$ if either $p1$ or $p2$ were selected as the next pivot. If $p1$ were the next pivot

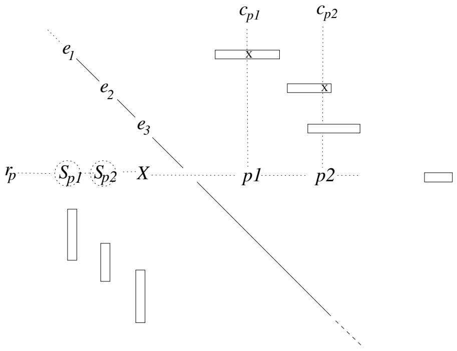


FIG. 3.2. *Influence of local symmetrization on the pivot structure.*

then the structure of row $r_p$ would be given by $\mathcal{U}_{p1} = \mathcal{A}_{r_p*} \cup \mathcal{U}_{e_3} \cup \mathcal{U}_{e_1}$ because of the locally symmetrized entry $S_{p1}$. If $p2$ were the next pivot then the structure of row $r_p$ would be given by $\mathcal{U}_{p2} = \mathcal{A}_{r_p*} \cup \mathcal{U}_{e_3} \cup \mathcal{U}_{e_2}$ because of the locally symmetrized entry $S_{p2}$. This shows that, even if we cannot anticipate the effect of local symmetrization on the quotient graph $\mathcal{G}_A$ before the pivot selection, we should anticipate its effect on the metrics used to select the best pivot between $p1$ and $p2$.

## 4. CMLS algorithm.

In this section, we describe the main features and properties of the `CMLS` algorithm. At each step of the algorithm, we need to know the exact structure of each row and column in $\mathbf{C}$. Moreover, we need to compute a metric that reflects the quality of each nonzero entry in $\mathbf{C}$. It is thus natural to use a bipartite graph (with possibly weighted edges) for $\mathbf{C}$. Each edge corresponding to a nonzero entry may have one or more weights that will be used to select a pivot. For example, a numerical value that approximates the magnitude of the entries and a structural metric that approximates the Markowitz cost (i.e., the product of the row and column degree) can be used. On the other hand, in order to have a fast computation of a structural metric based on the pattern of $\mathbf{A}$ and to have an in-place algorithm, $\mathbf{A}$ is represented by its quotient graph and local symmetrization is employed. The notation used to represent the quotient graph at each step of the algorithm is summarized in Table 4.1.

In section 4.1, we first describe the pivot selection algorithms. Updating the graphs $G_C$ and $\mathcal{G}_A$ associated with $\mathbf{C}$ and $\mathbf{A}$, respectively, is discussed in sections 4.2 and 4.3. In section 4.4 we describe how to compute, at each step $k$ and for each entry in the constraint matrix $\mathbf{C}^k$, structural metrics relative to $\mathcal{G}_A^k$. Section 4.5 finally explains how supervariables are defined and used in our context.

### 4.1. Pivot selection.

At each step, the best pivot according to a given metric is selected. The metric choice determines the underlying algorithmic strategy. We say that we use [illegible] in our algorithms when the entries are selected with respect to only information about the structure of the factors. In that case we will say that we use a [illegible]. When we combine a [illegible] metric and a [illegible] metric to select the pivot we will say that we use a [illegible].

Moreover, in sparse matrix factorization, we also want to preserve the sparsity of the factors while controlling the numerical growth in the factors. Numerical thresholds are introduced to give freedom for the pivot selection to balance numerical precision with sparsity preservation. An entry $(i, j) \in \mathbf{C}^k$ is said to be [illegible]

TABLE 4.1

*Notation used for bipartite quotient graph.*

| Bipartite quotient graph of **A**: $\mathcal{G}_A = (\mathcal{V}_r \cup \overline{\mathcal{V}}_r, \mathcal{V}_c \cup \overline{\mathcal{V}}_c, \mathcal{E} \cup \overline{\mathcal{E}})$ | |
|---|---|
| $\mathcal{R}_i$<br>$\mathcal{A}_{i*}$ | elements adjacent to row $i$<br>variables adjacent to row $i$ |
| $\mathcal{C}_j$<br>$\mathcal{A}_{*j}$ | elements adjacent to column $j$<br>variables adjacent to column $j$ |
| $\mathcal{U}_p$ | row structure of pivot $p$ after its elimination. |
| $\mathcal{L}_p$ | column structure of the pivot $p$ after its elimination. |
| $\mathcal{F}_p$ | elements that are adjacent to the row $r_p$ or the column $c_p$ of a noneliminated pivot $p$ ($\mathcal{F} = \mathcal{R}_{r_p} \cup \mathcal{C}_{c_p}$) |
| $\mathcal{F}$ | elements that are adjacent to row $i$ or column $j$ where $i$ and $j$ will depend on the context ($\mathcal{F} = \mathcal{R}_i \cup \mathcal{C}_j$) |

(or acceptable) according to a threshold $\tau$ if and only if $|c_{ij}| \geq \tau \times ||c_{.j}||_\infty$, where $\tau \in [0, 1]$. To reduce the complexity of the algorithms, it is also common to limit the pivot search to a set of candidate pivots. For example, in [13] the authors proposed to visit the entries of a fixed number of columns using the Zlatev-style search [37]. A similar strategy is used in [34] to find a pivot set in the context of parallel sparse **LU** factorization.

We use a slightly different algorithm: our pivot search is not restricted to columns but to a more complex set of entries in order to achieve a better fill-in reduction. At each step of the ordering, we look for the best entry $p = (r_p, c_p)$ within a subset (say, $S$) of the entries in the bipartite graph $G_C$. The subset $S$ is defined by two threshold parameters `MS` $> 0$ and `ncol` $\geq 0$ as follows. First, the `MS` entries with the smallest structural metric $m_0$ are added to $S$. Second, those `MS` entries may belong to several columns. We then add in $S$ all the other nonzero entries of those columns, but restricted to at most the first `ncol` columns. The set $S$ is thus composed of a first set of `MS` entries, the so-called `MS`-set, and a second set, the so-called `ncol`-set $= S \backslash$`MS`-set.

We now explain how we select the entry of minimum structural metric in $S$ among the numerically acceptable pivots. We first visit the `MS`-set sorted in increasing order of the structural metric $m_0$. The first numerically acceptable entry found corresponds to the minimum with respect to our hybrid strategy and we stop the search. Otherwise, none of the values in the `MS`-set entries is numerically acceptable. However, if `ncol` $> 0$ then we are sure that at least `ncol` entries will be numerically acceptable since $\tau \leq 1$. Finally, if `ncol` $= 0$ and none of the entries in the `MS`-set is numerically acceptable then the first entry of the `MS`-set is selected even if it is not numerically acceptable. (In our experiments `MS` $= 100$ and `ncol` $= 10$ are used.)

### 4.2. Update of the bipartite graph $G_C$.

A bipartite graph is used to represent **C**. At each step $k$, we need to add new entries in $G_{C^{k+1}}$ corresponding to the fill-ins in $\mathbf{C}^{k+1}$. Since $G_C$ holds the set of candidate pivots, we need to guarantee that properties (2.1) and (2.2) hold.

Let $\mathcal{M}$ be a matching in $\mathbf{C}^k$. The following two extreme strategies preserve these two properties:

- `MATCH_UPDATE` will refer to the strategy that performs incomplete Gaussian elimination on **C** to preserve only the perfect matching property (2.1). Let $p = (r_p, c_p)$ be the current pivot. Let ($r_p$, `match_col`) and (`match_row`, $c_p$) be the matched entries of **C** in row $r_p$ and column $c_p$, respectively. That is, ($r_p$, `match_col`) $\in \mathcal{M}$ and (`match_row`, $c_p$) $\in \mathcal{M}$. If these entries are the same (i.e., $(r_p, c_p)$ is a matched entry), nothing needs to be done to maintain

property (2.1). Otherwise, entry (`match_row`, `match_col`) is added to $\mathbf{C}$ and $\mathcal{M}$ to maintain property (2.1). Note that this entry corresponds to an entry in $\mathcal{P}attern(\bar{\mathbf{C}}^k)$, so that property (2.2) remains true.
- `TOTAL_UPDATE` will refer to the strategy which performs all the updates in $\mathbf{C}$ (i.e., $\mathbf{C}^{k+1} = \bar{\mathbf{C}}^k$). Note that even if this strategy naturally preserves property (2.2), our perfect matching on $\mathbf{C}^{k+1}$ may have to be updated as in the `MATCH_UPDATE` strategy.

In practice, a mixed strategy, exploiting both `MATCH_UPDATE` and `TOTAL_UPDATE`, will be used for the experiments. (The decision is based on memory and cost estimations of the algorithm.)

**4.3. Update of the bipartite quotient graph $\mathcal{G}_A$.** In Algorithm 4.1 we describe how the bipartite quotient graph associated with the reduced matrix is updated.

**Algorithm 4.1** `CMLS` update of the bipartite quotient graph $\mathcal{G}_A^k$

```
   Let $p = (r_p, c_p)$ be the current pivot at step $k$ and $\mathcal{F}_p = \mathcal{R}_{r_p} \cup \mathcal{C}_{c_p}$.
   if $\mathcal{U}_p \neq \emptyset$ and $\mathcal{L}_p \neq \emptyset$ then
     for each row $i \in \mathcal{L}_p$ do
1      $\mathcal{A}_{i*} = (\mathcal{A}_{i*} \setminus \mathcal{U}_p) \setminus \{c_p\}$ /* variable elimination in row direction */
2      $\mathcal{R}_i = (\mathcal{R}_i \setminus \mathcal{F}_p) \cup p$
     end for
     for each column $j \in \mathcal{U}_p$ do
3      $\mathcal{A}_{*j} = (\mathcal{A}_{*j} \setminus \mathcal{L}_p) \setminus \{r_p\}$ /* variable elimination in column direction */
4      $\mathcal{C}_j = (\mathcal{C}_j \setminus \mathcal{F}_p) \cup p$
     end for
   else /* pivot pruning: delete all that is related to $p$, if $\mathcal{U}_p = \emptyset$ or $\mathcal{L}_p = \emptyset$ */
     for each row $i \in \mathcal{L}_p$ do
       $\mathcal{R}_i = (\mathcal{R}_i \setminus \mathcal{F}_p)$
       $\mathcal{A}_{i*} = \mathcal{A}_{i*} \setminus \{c_p\}$
     end for
     for each column $j \in \mathcal{U}_p$ do
       $\mathcal{C}_j = (\mathcal{C}_j \setminus \mathcal{F}_p)$
       $\mathcal{A}_{*j} = \mathcal{A}_{*j} \setminus \{r_p\}$
     end for
   end if
```

The "if" block of Algorithm 4.1 shows how the elements and variables are pruned. The element pruning performed at lines 2 and 4 includes pruning due to local symmetrization. The variable pruning performed at lines 1 and 3 removes the intersection of the adjacency structures. For each row $i$ in $\mathcal{L}_p$, variables of $\mathcal{A}_{i*}$ that appear in $\mathcal{U}_p$ are removed and we say that we perform [illegible]. For each column $j$ in $\mathcal{U}_p$, variables of $\mathcal{A}_{*j}$ that appear in $\mathcal{L}_p$ are removed. This will be referred to as [illegible]. We then say that our algorithm performs [illegible]. Note that if, at a given step, variables are removed from both row $i$ and column $i$, it means that $i \in \mathcal{L}_p$ and $i \in \mathcal{U}_p$. In section 4.3.1, we will prove that under additional assumptions more pruning of the variables could have been introduced. We then however comment in section 4.3.2 that doing so makes impossible the detection of the reducibility as done in the "else" block of Algorithm 4.1. We will also explain why it is correlated with the strategy used to prune variables. Note that this additional pruning would have improved the accuracy of our structural metrics as explained in section 4.4.

**4.3.1. Two-way variable elimination.** Property 4.1 shows that under additional assumptions the structure of the quotient graph can be further pruned.

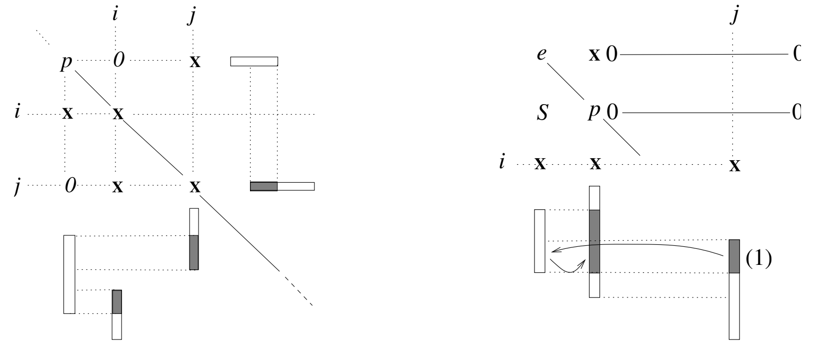
PROPERTY 4.1. [illegible] $k$ [illegible] $(i,j)$ [illegible] $i$ [illegible] $j$ [illegible] $\mathbf{C}^k$ [illegible]

(1) [illegible] $i \in \mathcal{L}_p$ [illegible] $\mathcal{L}_p$ [illegible] $\mathcal{A}_{*j}$ [illegible] $j \notin \mathcal{U}_p$

(2) [illegible] $j \in \mathcal{U}_p$ [illegible] $\mathcal{U}_p$ [illegible] $\mathcal{A}_{i*}$ [illegible] $i \notin \mathcal{L}_p$

[illegible]. From (2.2), we have $\mathcal{P}attern(\mathbf{C}^{k+1}) \subseteq \mathcal{P}attern(\bar{\mathbf{C}}^k)$. Therefore, if at step $k$, $(i,j)$ is the only entry in row $i$ and column $j$ of $\mathbf{C}^k$, it will remain the only entry in its row and column for all subsequent $\mathbf{C}^l$ for $l > k$. Thus $(i,j)$ will be selected as a pivot in a future step, and we can anticipate where local symmetrization will occur. So the entries in $\mathcal{A}_{*j} \cap \mathcal{L}_p$ for Property 4.1(1) (or in $\mathcal{A}_{i*} \cap \mathcal{U}_p$ for Property 4.1(2)) can be pruned and will be retrieved from $\mathcal{L}_p$ (or $\mathcal{U}_p$) when $(i,j)$ is eliminated. ☐

When we apply Property 4.1, we say that the algorithm performs elimination in both row and column directions. This process will be referred to as [illegible]. For example, when the pivot choice is limited to a transversal, the two-way variable elimination can be performed at each step of the elimination, as in the `DMLS` algorithm [5]. This is illustrated in Figure 4.1(a). We assume, for the sake of clarity, that the input matrix has been permuted to have all the candidate pivots on the diagonal. The shaded areas correspond to the variables that can be removed from the variable adjacency lists because they are implicitly stored through the adjacency lists of element $p$.



(a) Illustration of two-way variable elimination.

(b) Effect of variable elimination in both directions on reducibility detection (S indicates the position of local symmetrization).

FIG. 4.1. *Variable elimination and reducibility detection.*

If the hypothesis of Property 4.1 is not true, the two-way variable elimination cannot be applied because we do not know whether local symmetrization will be performed or not. Let us consider Figure 4.1(a) again. If all three entries $(i,i)$, $(j,j)$, and $(j,i)$ belong to $\mathbf{C}$, then we cannot prune all the shaded areas. This is because both $(i,i)$ and $(j,i)$ are potential pivots from column $i$. If $(i,i)$ were chosen as the pivot from column $i$, then the shaded area in column $i$ could have been pruned during the elimination of $p$ thanks to local symmetrization relative to entry $i$ in column $\mathcal{L}_p$. However, if $(j,i)$ were selected as the pivot from column $i$, then since $j \notin \mathcal{L}_p$ and $i \notin \mathcal{U}_p$, the element $p$ would not be used to build the row and column adjacency of $(j,i)$. In this case the shaded area in column $i$ should not be pruned during the elimination of $p$ since it would be impossible to retrieve those variables. Note that the shaded area in column $j$ can be pruned, because the entry $(p, j)$ is [illegible], and so the variable elimination in the column direction can be applied.

**4.3.2. Reducibility detection.** If the input matrix is reducible, we may encounter a pivot $p$ such that either (1) both $\mathcal{L}_p = \emptyset$ and $\mathcal{U}_p = \emptyset$ (referred to as [illegible]) or (2) $\mathcal{L}_p = \emptyset$ or $\mathcal{U}_p = \emptyset$ (referred to as [illegible]). Ideally, we would like to remove $p$ from the quotient graph $\mathcal{G}_A$ in both reducible cases. (In our context [illegible] from $\mathcal{G}_A$ means that both the pivot and its adjacency structure can be suppressed without any further update of the graph.) However, we will show that whether $p$ can be removed or not depends on whether we use only one-way variable elimination or we use two-way variable elimination as well.

PROPERTY 4.2. [illegible] $p$ [illegible]

Property 4.2 comes from the fact that the pruning of the structures due to local symmetrization has not been anticipated. Thus, none of the entries in $\mathcal{L}_p$ (if $\mathcal{U}_p = \emptyset$) or $\mathcal{U}_p$ (if $\mathcal{L}_p = \emptyset$) will be needed by the other variables to represent their adjacency structure in $\mathcal{G}_A$. Therefore, pivot $p$ can be removed from $\mathcal{G}_A$.

PROPERTY 4.3. [illegible] $p$ [illegible]

[illegible]. First, when $\mathcal{U}_p = \emptyset$ and $\mathcal{L}_p = \emptyset$, $p$ becomes a singleton element and can certainly be removed from the quotient graph. Second, let us suppose that two-way variable elimination has been performed at least once. We now build a counterexample to show that we cannot safely (in all possible cases) remove a weakly reducible pivot $p$. Let us assume without loss of generality that $\mathcal{U}_p = \emptyset$ and $\mathcal{L}_p \neq \emptyset$. Let us assume that there exists a variable $i \in \mathcal{L}_p$ and that there is an entry $(i, j)$ that is the only entry in row $i$ and column $j$ in $\mathbf{C}$. We also assume that variables in $\mathcal{A}_{*j}$ have been pruned under two-way variable elimination. Therefore, $p$ must be used to retrieve those entries and cannot be removed from $\mathcal{G}_A$. This is illustrated in Figure 4.1(b) where the shaded area (1) in column $j$ is first stored through element $e$ and then stored through element $p$ (after pivot $p$ absorbs element $e$). ☐

Property 4.3 indicates a drawback of the two-way variable elimination: we can only prune the pivot in the strongly reducible case. The algorithm may be very inefficient if the matrix is very reducible in the weak sense.

When the matrix is reducible to block triangular form (BTF), the reducibility detection may have a significant impact on the ordering quality [15]. In that case many instances in which an element is strongly or weakly reducible can appear. Property 4.2 can then be used to show that thanks to one-way variable elimination `CMLS` will better detect and exploit the BTF of a matrix (see [6, 32] for further details).

**4.4. Update of the structural metric.** In this section, we describe heuristics to estimate the structural quality of a pivot.

In the preamble section, we first describe how we approximate the row and column degrees. In section 4.4.2, we describe a metric based on an upper bound on the fill-ins introduced at each step of elimination. This approximation of the fill-ins has been studied by the authors of `AMD` [1] for symmetric matrices. We provide a generalization of this approximation to unsymmetric matrices and prove that it is a tighter upper bound on the fill-ins than the approximations proposed for symmetric matrices in [33]. Note that concerning the deficiency approximation in [30], there is no guarantee that it is an upper bound of the fill-in. Our approximate minimum fill-in heuristic will be referred to as `AMFI`.

**4.4.1. Preamble.** Let us assume that the $k$th pivot $p = (r_p, c_p)$ has been selected. All the entries in $(\mathcal{L}_p \times V_c \;\cup\; V_r \times \mathcal{U}_p) \cap \mathcal{P}attern(\mathbf{C})$ are involved in the structural metric updates. The size of this area is thus larger than the area involved in the update of the structure of $\mathbf{C}$. The algorithm to update the structural metrics is one of the most costly steps of our algorithm.

We want the metrics to reflect the structural quality of an entry if it were selected as the next pivot. That is why we compute metrics which are related to the structure of our quotient graph and for which local symmetrization is anticipated. In the following, the degrees, approximate degrees, fill-ins, and approximate fill-ins are all related to this quotient graph structure.

Let $d_r(i,j)$ and $d_c(i,j)$ denote, respectively, the external row and the external column degrees of entry $(i,j)$. Similarly to the `AMD` [1] and `DMLS` [5] algorithms, approximate row and column external degrees are computed. The `AMD`-like approximate external row and column degrees, $amd_r(i,j) > d_r(i,j)$ and $amd_c(i,j) > d_c(i,j)$, respectively, are then defined by the following two equations:

$$\begin{aligned} amd_r(i,j) = \;& |\mathcal{A}_{i*} \setminus \mathcal{U}_p| + |\mathcal{U}_p \setminus j| + \textstyle\sum_{e \in \mathcal{R}_i \cup \mathcal{C}_j} (|\mathcal{U}_e \setminus \mathcal{U}_p|) - \alpha_j, \\ & \text{with } \alpha_j = \max(|\mathcal{C}_j|, 1) \text{ if } j \notin \mathcal{U}_p \text{ else } \alpha_j = 0. \end{aligned} \tag{4.1}$$

$$\begin{aligned} amd_c(i,j) = \;& |\mathcal{A}_{*j} \setminus \mathcal{L}_p| + |\mathcal{L}_p \setminus i| + \textstyle\sum_{e \in \mathcal{C}_j \cup \mathcal{R}_i} (|\mathcal{L}_e \setminus \mathcal{L}_p|) - \beta_i, \\ & \text{with } \beta_i = \max(|\mathcal{R}_i|, 1) \text{ if } i \notin \mathcal{L}_p \text{ else } \beta_i = 0. \end{aligned} \tag{4.2}$$

As was done in [5], degree corrections ($\alpha_j$ and $\beta_i$ in (4.1) and (4.2)) are introduced to improve the approximations of the row and column external degrees in the presence of local symmetrization. To justify these correction terms, one can observe that if $j \notin \mathcal{U}_p$ then $j$ is counted in every $\mathcal{U}_e \setminus \mathcal{U}_p$ for $e$ that is adjacent to column $j$ ($e \in \mathcal{C}_j$). Furthermore, if $\mathcal{C}_j$ is empty and $j \notin \mathcal{U}_p$ then column $j$ has been counted in $\mathcal{A}_{i*} \setminus \mathcal{U}_p$ and should then be subtracted. This explains the use of $\alpha_j$ in the correction. $\beta_i$ can be justified in a similar way. The $|\mathcal{U}_e \setminus \mathcal{U}_p|$ and $|\mathcal{L}_e \setminus \mathcal{L}_p|$ quantities are computed similarly in the `AMD` algorithm.

Note that since only one-way variable elimination is employed, the computation of the metric is less accurate than with two-way variable elimination. This is because in the latter case, for any element $e$, row index $i \in \mathcal{U}_p$, and column index $j \in \mathcal{L}_p$, we have $\mathcal{A}_{i*} \cap \mathcal{U}_e = \emptyset$ and $\mathcal{A}_{*j} \cap \mathcal{L}_e = \emptyset$. This is no longer true when one-way variable elimination is used (see Algorithm 4.1). But as was explained in section 4.3, the benefit of one-way variable elimination is to better exploit the BTF of the matrix.

After eliminating the $k$th pivot, we approximate the row and column degree by

$$\overline{amd}_r(i,j) = \min(amd_r(i,j), n-k-1) \tag{4.3}$$

and

$$\overline{amd}_c(i,j) = \min(amd_c(i,j), n-k-1). \tag{4.4}$$

Note that these approximations do not use the values of the previous approximate row and column degrees because it would be costly to store these quantities for each entry in $\mathbf{C}$.

**4.4.2. Approximation of the fill-in.** We want to estimate the amount of new fill-in that would occur in the reduced matrix if an entry were selected as the next pivot. A coarse upper bound of the fill-in that would occur can be obtained by removing the area corresponding to $\mathcal{L}_p \times \mathcal{U}_p$ from the Markowitz cost or the area corresponding to the largest adjacent clique [33]. A tighter approximation of the fill-in in the factors can be obtained by removing all the areas already filled during the elimination of the previous elements.

Suppose that $i \in \mathcal{L}_p$ or $j \in \mathcal{U}_p$. Let $\mathcal{F} = \mathcal{R}_i \cup \mathcal{C}_j$. Let $e$ be an element that belongs to $\mathcal{F}$. Let $S(i, j)$ denote the union of the areas associated with all the elements adjacent to entry $(i, j)$:

$$S(i,j) = \left| \bigcup_{e \in \mathcal{F}} (\mathcal{L}_e \setminus \{i\}) \times (\mathcal{U}_e \setminus \{j\}) \setminus (\mathcal{L}_p \times \mathcal{U}_p) \right| .$$

Ideally one might want to subtract both $\mid (\mathcal{L}_p \setminus \{i\}) \times (\mathcal{U}_p \setminus \{j\}) \mid$ and $S(i, j)$ from the Markowitz cost $d_r(i, j) \times d_c(i, j)$. An upper bound of the fill-in that would occur (including local symmetrization) if an entry $(i, j)$ were eliminated is

$$d_r(i,j) d_c(i,j) - \mid (\mathcal{U}_p \setminus \{j\}) \mid \mid (\mathcal{L}_p \setminus \{i\}) \mid - S(i,j).$$

The authors of [33] have observed that instead of using the exact external degrees one could use the approximate (in the sense of the `AMD` algorithm) external degrees since both produce results of comparable quality and since `AMD`-based metrics are significantly faster to compute. In this context, the corresponding upper bound of the fill-in metric becomes

$$amd_r(i,j) amd_c(i,j) - \mid (\mathcal{U}_p \setminus \{j\}) \mid \mid (\mathcal{L}_p \setminus \{i\}) \mid - S(i,j). \tag{4.5}$$

Let $AS$ be an overestimation of area $S$,

$$AS(i,j) = \sum_{e \in \mathcal{F}} \mid (\mathcal{L}_e \setminus \{i\}) \times (\mathcal{U}_e \setminus \{j\}) \setminus (\mathcal{L}_p \times \mathcal{U}_p)|. \tag{4.6}$$

Property 4.4 proves that one can in fact subtract area $AS(i, j)$, instead of $S(i, j)$, to obtain a more accurate upper bound of the fill-in metric than expression (4.5).

PROPERTY 4.4. $amd_r(i,j) amd_c(i,j) - \mid (\mathcal{U}_p \setminus \{j\}) \mid \mid (\mathcal{L}_p \setminus \{i\}) \mid - AS(i,j)$ [illegible] $(i, j)$ [illegible]

An intuitive proof of Property 4.4 is that, during the computation of the approximate degree, the submatrix is expanded in such a way that the intersections between all $(\mathcal{U}_e \setminus \mathcal{U}_p \setminus \{j\})$ and between all $(\mathcal{L}_e \setminus \mathcal{L}_p \setminus \{i\})$ for $e \in \mathcal{F}$ are empty. The area $AS$ corresponds to a real surface in the expanded matrix and can be removed from the area $amd_r(i, j) amd_c(i, j)$ to compute the fill-in that would occur in the expanded matrix. Moreover, this fill-in in the expanded matrix is an upper bound of the exact fill-in in the quotient graph. A formal proof of Property 4.4 is given in [32].

In practice we use $\overline{amd}_r(i, j)$ and $\overline{amd}_c(i, j)$ as defined in (4.3) and (4.4) instead of $amd_r(i, j)$ and $amd_c(i, j)$. Because of that it may happen that $\overline{amd}_r(i,j) \overline{amd}_c(i,j) - |\mathcal{U}_p \setminus j| |\mathcal{L}_p \setminus i| - AS(i,j)$ becomes negative, meaning that either $\overline{amd}_r(i,j) < amd_r(i,j)$ or $\overline{amd}_c(i,j) < amd_c(i,j)$. In such cases, as it is done in `AMD` and `DMLS`, one can artificially set the metric to 0. We propose here an alternative that could also be applied to these approaches to limit the tie-breaking. We introduce row and column scaling terms

$$\texttt{rowscale} = \frac{\overline{amd}_r(i,j) - |\mathcal{U}_p \setminus j|}{amd_r(i,j) - |\mathcal{U}_p \setminus j|} \text{ and } \texttt{colscale} = \frac{\overline{amd}_c(i,j) - |\mathcal{L}_p \setminus i|}{amd_c(i,j) - |\mathcal{L}_p \setminus i|}.$$

If one systematically scales the area $AS$ by `rowscale` × `colscale`, then we ensure a positive metric and avoid tie-breaking problems due to metrics equal to 0. Our final `AMFI` metric is then defined as follows:

$$metric^{(k+1)}(i,j) = \min \left\{ \begin{array}{ll} \multicolumn{2}{l}{\overline{amd}_r(i,j)\overline{amd}_c(i,j) - |\mathcal{U}_p \setminus j||\mathcal{L}_p \setminus i|} \\ \multicolumn{2}{l}{\quad - \texttt{rowscale} \times \texttt{colscale} \times AS(i,j)} \\ metric^{(k)}(i,j) & +|\mathcal{U}_p \setminus j| \times \overline{amd}_r(i,j) \\ & +|\mathcal{L}_p \setminus i| \times \overline{amd}_c(i,j) \\ & -2 \times |\mathcal{U}_p \setminus j| \times |\mathcal{L}_p \setminus i|. \end{array} \right. \tag{4.7}$$

**4.5. Supervariables and mass elimination.** For the sake of clarity, the algorithms described in the previous section did not include supervariables. In this section, we first define our generalization of [illegible] and [illegible] to bipartite quotient graphs with off-diagonal pivots. We then revisit the previous algorithms and explain what has to be modified to detect and exploit supervariables.

In our context, we want supervariables to exploit identical adjacency structures in the graph at each step of the elimination. Supervariables are thus defined on the bipartite quotient graph of **A**, whereas on the bipartite graph of **C** we use only simple variables. With the `CMLS` algorithm we cannot use exactly the same kinds of supervariables as in [1, 5, 19, 23] because they assume that pivots are on the diagonal so that a row can be associated with a column before being selected as a pivot. That is why our concept of supervariable is closer to the one used in [22]: we define [illegible] (resp., [illegible]) as row variables (resp., columns variables) which have the same adjacency in $\mathcal{G}_A$. To limit the cost of supervariable detection, two hash functions (see, for example, [9]) are then used for each row and column direction.

If $i$ and $j$ are two indistinguishable row variables, they are replaced in $\mathcal{G}_A$ by a [illegible] containing both $i$ and $j$, labeled by its [illegible] ($i$, say) [18, 19, 20]. The notation $\mathbf{i}$ is used to denote this row supervariable and $\mathbf{i} = \{i, j\}$. $i$ and $j$ are said to be [illegible] of the row supervariable $\mathbf{i}$ and the notations $i \in \mathbf{i}$ and $j \in \mathbf{i}$ are then used. At the beginning of Gaussian elimination, the row variables are said to be [illegible]. Each simple row variable $i$ can also be seen as a row supervariable $\mathbf{i} = \{i\}$. For each row supervariable $\mathbf{i}$, $|\mathbf{i}|$ corresponds to its size, i.e., its number of constituent variables. Similar definitions and notation can be introduced for the [illegible], the [illegible], the [illegible], and the [illegible]. When it is clear from the context, we do not differentiate between a column or a row supervariable. Furthermore, let $r_1$ and $r_2$ be two row variables which belong to the same row supervariable $\mathbf{r}$ and $c_1$ and $c_2$ be two column variables which belong to the same column supervariable $\mathbf{c}$. After the elimination of pivot $p_1 = (r_1, c_1)$, $p_2 = (r_2, c_2)$ can be eliminated in $\mathcal{G}_A$ without causing extra fill-in. This process, commonly referred to as [illegible] [25], creates a new (super)element $e = (\mathbf{r}_e, \mathbf{c}_e)$ in the quotient graph with $\mathbf{r}_e = \{r_1, r_2\}$ and $\mathbf{c}_e = \{c_1, c_2\}$. In the following, we comment on the algorithmic modifications due to the introduction of supervariables.

Let $p$ be the current pivot. The first modification of the algorithm concerns the introduction of a scaling of the structural metric as defined by (4.7). The structural metric of an entry $(\mathbf{i},\mathbf{j})$ adjacent to $p$ either in the row or in the column direction is divided by $\min(|\mathbf{i}|,|\mathbf{j}|)$. Indeed, $\min(|\mathbf{i}|,|\mathbf{j}|)$ corresponds to the size of the largest pivot block which could be eliminated if a pivot at the intersection of these row and column supervariables were selected.

The second modification of the algorithm concerns the elimination process which is performed in the following three main steps. During the first step, the scaled metric is used to select a pivot in $\mathbf{C}$. During the second step, we retrieve its associated row $\mathbf{r}_p$ and column $\mathbf{c}_p$ supervariables in $\mathbf{A}$. During the third step, we eliminate "as many as possible" variables belonging to $(\mathbf{r}_p \times \mathbf{c}_p) \cap \mathbf{C}$. Note that the meaning of "as many as possible" will depend on the context. If a hybrid strategy is used then pivot entries might be rejected because of numerical criteria. Furthermore, since the $\mathbf{C}$ matrix is updated when eliminating a pivot, the new nonzero entries that might be at the intersection of the pattern of $\mathbf{C}$ and the supervariables need also be considered. The same modified three steps are also applied when the mass elimination process of supervariables adjacent to the current pivot is involved. Finally, if some constituent variables of a supervariable have not been eliminated, then they are used to build a new supervariable and are reinserted in $\mathcal{G}_A$.

The final modification concerns the update of the structural metric. After the elimination of a pivot $p$, the approximate external row and column degrees as defined by (4.1) and (4.2) become

$$\begin{aligned} amd_r(i,j) = \; & |\mathcal{A}_{\mathbf{i}*} \setminus \mathcal{U}_p| + |\mathcal{U}_p \setminus \{\mathbf{j}\}| + \textstyle\sum_{e\in\mathcal{R}_{\mathbf{i}}\cup\mathcal{C}_{\mathbf{j}}}(|\mathcal{U}_e \setminus \mathcal{U}_p|) - \alpha_{\mathbf{j}}\,|\mathbf{j}|, \\ & \text{with } \alpha_{\mathbf{j}} = \max(|\mathcal{C}_{\mathbf{j}}|,1) \text{ if } \mathbf{j} \notin \mathcal{U}_p \text{ else } \alpha_{\mathbf{j}} = 0, \end{aligned} \tag{4.8}$$

$$\begin{aligned} amd_c(i,j) = \; & |\mathcal{A}_{*\mathbf{j}} \setminus \mathcal{L}_p| + |\mathcal{L}_p \setminus \{\mathbf{i}\}| + \textstyle\sum_{e\in\mathcal{R}_{\mathbf{i}}\cup\mathcal{C}_{\mathbf{j}}}(|\mathcal{L}_e \setminus \mathcal{L}_p|) - \beta_{\mathbf{i}}\,|\mathbf{i}|, \\ & \text{with } \beta_{\mathbf{i}} = \max(|\mathcal{R}_{\mathbf{i}}|,1) \text{ if } \mathbf{i} \notin \mathcal{L}_p \text{ else } \beta_{\mathbf{i}} = 0. \end{aligned} \tag{4.9}$$

**5. Experiments.** In this section we analyze the effect of the `CMLS` ordering on the performance of sparse solvers. Our new ordering will be compared to the combination of `DMLS` ordering and `MC64` [16, 17] because it is the most robust in-place local heuristic (better than the combination of `AMD` and `MC64`; see [5]) in terms of numerical stability and fill-in reduction in the factors. `DMLS` takes into account the asymmetry of the matrices, selects pivots on the diagonal, and applies local symmetrization and two-way variable elimination. Thus it can be considered a restricted `CMLS`. We recall that `MC64` permutes the matrix such that the product of the diagonal elements is maximized.

With the `CMLS` ordering, our pivot sequence results from a combination of structural and numerical information (even when only structural metrics are used to select the pivots, the initialization of our constraint matrix is based on numerical considerations). Therefore it is important to analyze the numerical quality of the proposed sequence of pivots. In this context, for very different motivations, we may want to experiment with both an approach that performs partial pivoting to preserve numerical stability and an approach based on static pivoting. In the first case, the numerical quality of the proposed sequence of pivots is not so critical to obtaining a backward stable factorization, and we expect to improve the sparsity of the factors because of the freedom to select entries in the constraint matrix $\mathbf{C}$. In the case of a static pivoting, we expect that the capacity of `CMLS` to select pivots according to numerical criteria can be used to better control the numerical quality of the sequence while still offering more freedom than a diagonal Markowitz algorithm. In fact with the `CMLS` algorithm we can define a family of orderings and expect that two probably different members of this family can be used in these two cases: a `CMLS` ordering in which $\mathbf{C}$ offers a lot of freedom to choose the pivots, and a `CMLS` ordering in which the selection of the pivots is strongly guided by the numerical values in $\mathbf{C}$.

To represent each class of solver techniques, we consider the multifrontal code `MA41_UNS` [2, 7] which performs numerical pivoting during the factorization and the supernodal code `SuperLU_DIST` [28] which performs static pivoting. Both codes are run in sequential mode. As shown in [4, 7, 12, 26] the approaches used to factorize the matrix in `MA41_UNS` and `SuperLU_DIST` are very competitive in shared/sequential and distributed memory environments, respectively. Note that because of the important algorithmic similarities between `MA41_UNS` and the distributed memory code `MUMPS` [3], this work also will be very beneficial to the distributed memory multifrontal code.

In section 5.1, we present our experimental environment. In section 5.2, we discuss the case where the pivot choice in `CMLS` is restricted to the matching provided by `MC64`. In section 5.3, we analyze the behavior of our ordering when a structural strategy is used to select the pivots. We report performance obtained with `MA41_UNS` in terms of time and memory used during factorization. In section 5.4, we illustrate the benefits resulting from the use of hybrid strategies for pivot selection in `SuperLU_DIST` and focus on the numerical effects.

### 5.1. Experimental environment.

**5.1.1. Test matrices and computing environment.** Consider a matrix $\mathbf{A} = (a_{ij})$ and let $nnz(\mathbf{A})$ be its number of nonzero entries. We define the [illegible] [illegible] $s(\mathbf{A})$ as

$$s(\mathbf{A}) = \frac{|\{(i,j) \text{ s.t. } a_{ij} \neq 0 \text{ and } a_{ji} \neq 0\}|}{nnz(\mathbf{A})}.$$

If $\mathbf{A}$ is symmetric, then $s(\mathbf{A}) = 1$, and if $\mathbf{A}$ is strictly triangular, then $s(\mathbf{A}) = 0$. In the remainder of this section, the symmetry of a matrix always refers to the structural symmetry after the `MC64` permutation has been applied (see column sym of Table 5.1).

A representative set of 19 large unsymmetric matrices has been selected from Davis's collection [11]; see Table 5.1. Only matrices with a structural symmetry lower than 0.5 and of order greater than 10000 were chosen. Moreover, we limited the number of similar matrices from the same family to two in order to avoid the class effects. We also added to our test set four matrices (mixtank, invextr1, fidapm11, and cavity16) from the PARASOL test data[1] and Matrix Market,[2] because we have observed that for these matrices `SuperLU_DIST` needs iterative refinement to improve the accuracy of the solution [4]. These four matrices will be used in section 5.4 to illustrate that using the `CMLS` ordering improves the numerical behavior of `SuperLU_DIST`.

All our results were obtained on a Linux PC computer (Pentium 4, 2.8 GHz, 2 GBytes of memory, and 1 MByte of cache). We used the Portland Fortran 90 compiler `pgf90`, C compiler `gcc` (both with -O3 option), and ATLAS BLAS [35, 36].

We systematically applied random row and column permutations to our initial matrix so that the ordering algorithms were less sensitive to the effects of tie-breaking. We ran each problem with eleven random permutations and selected the run whose

---

[1] http://www.parallab.uib.no/projects/parasol/data

[2] http://math.nist.gov/MatrixMarket

TABLE 5.1
*Test matrices.*

| Group/Matrix | n | nnz | sym | Description |
|---|---:|---:|---:|---|
| Vavasis/av41092 | 41092 | 1683902 | 0.08 | Unstructured finite element |
| Hollinger/g7jac200sc | 59310 | 837936 | 0.10 | Economic model |
| Hollinger/g7jac180sc | 53370 | 747276 | 0.10 | Economic model |
| Hollinger/jan99jac120sc | 41374 | 260202 | 0.16 | Economic model |
| Hollinger/jan99jac100sc | 34454 | 215862 | 0.16 | Economic model |
| Mallya/lhr34c | 35152 | 764014 | 0.19 | Light hydrocarbon recovery |
| Mallya/lhr71c | 70304 | 1528092 | 0.20 | Light hydrocarbon recovery |
| Hollinger/mark3jac120sc | 54929 | 342475 | 0.21 | Economic model |
| Hollinger/mark3jac140sc | 64089 | 399735 | 0.21 | Economic model |
| Grund/bayer01 | 57735 | 277774 | 0.25 | Chemical process simulation |
| Hohn/sinc18 | 16428 | 973826 | 0.27 | Single-material crack problem (sinc-basis) |
| Hohn/sinc15 | 11532 | 568526 | 0.27 | Single-material crack problem (sinc-basis) |
| Zhao/Zhao2 | 33861 | 166453 | 0.27 | Electromagnetism |
| Sandia/mult_dcop_03 | 25187 | 193216 | 0.36 | Circuit simulation |
| ATandT/twotone | 120750 | 1224224 | 0.42 | Harmonic balance method |
| ATandT/onetone1 | 36057 | 341088 | 0.42 | Harmonic balance method |
| Norris/torso1 | 116158 | 8516500 | 0.43 | Finite element matrices from bioengineering |
| Simon/bbmat | 38744 | 1771722 | 0.49 | 2D airfoil, turbulence |
| Shen/shermanACb | 18510 | 145149 | 0.50 | Matrices from Kai Shen |
| mixtank | 29957 | 1995041 | 0.91 | fluid flow (PARASOL, Polyflow S.A.) |
| invextr1 | 30412 | 1793881 | 0.85 | fluid flow (PARASOL, Polyflow S.A.) |
| fidapm11 | 22294 | 623554 | 0.45 | CFD (SPARSKIT2 collection) |
| cavity16 | 4562 | 138187 | 0.84 | Finite element modeling (SPARSKIT2 collection) |

ordering returns the median fill-in in the factors. We did not observe large variations of the amount of fill-in in the factors from these random permutations except for matrix bbmat. (In general variations are smaller than 10%.)

**5.1.2. CMLS and DMLS testing environment.** The initialization of $\mathbf{C}$ is done using a scaled matrix and the maximum weighted matching returned by `MC64`. To limit the size of $\mathbf{C}$ and the complexity (cost and memory) of the ordering phase, the initial number of entries in $\mathbf{C}^0$ is set between $n$ and $4n$ (computation based on a function that depends on both $n$ and $nnz(\mathbf{A})$). We then drop the entries that are smaller than 0.1 in magnitude and the entries whose structural metrics are too large. While dropping, we still maintain the nonsingularity property (2.1). In our test set, we observed that the size of $\mathbf{C}^0$ is between $n$ and $3n$ after this last dropping phase.

We use the metric `AMFI` of section 4.4.2 since it is the most efficient metric for both `CMLS` and `DMLS` orderings. In the `CMLS` implementation, we use `rowscale` and `colscale` coefficients (see end of section 4.4.2) to reduce the amount of tie-breaking between variables that would have a negative metric (reset to 0) with `DMLS`. This algorithmic modification has also been implemented in the `DMLS` code to simplify our discussions in this section.

**5.2. Preliminary remarks about diagonal constraint matrix.** When $\mathbf{C}^0$ contains only the entries from the `MC64` matching and thus the set of candidate pivots for `CMLS` and `DMLS` is identical, one should expect a comparable behavior of the two algorithms in terms of fill-in in the factors. However, we have noticed that `CMLS` ordering tends to produce sparser factors (see [32] for detailed results) even if `DMLS` uses two-way variable elimination which leads to more accurate structural metrics, as explained in section 4.4.1. This can be explained by the following algorithmic differences:

Table 5.2

`MA41_UNS` *size of the factors and analysis reliability. Each number for the factor size is in thousands.* `std`*: Standard deviation over the eleven runs. Mean (resp., Median): Mean (resp., median) value of the ratio* `CMLS` *statistic /* `DMLS` *statistic.*

| | Estimated size of factors | | | | Real size of factors | | Ratio: Actual/predicted | |
|---|---|---|---|---|---|---|---|---|
| Matrix | `CMLS` | `std` | `DMLS` | `std` | `CMLS` | `DMLS` | `CMLS` | `DMLS` |
| av41092 | 6609 | 0.01 | 9323 | 0.02 | 6849 | 9553 | 1.03 | 1.02 |
| g7jac200sc | 27912 | 0.06 | 30424 | 0.02 | 28282 | 30443 | 1.01 | 1.00 |
| g7jac180sc | 24755 | 0.04 | 26789 | 0.03 | 25077 | 26810 | 1.01 | 1.00 |
| jan99jac120sc | 3191 | 0.02 | 4326 | 0.03 | 3197 | 4330 | 1.00 | 1.00 |
| jan99jac100sc | 2651 | 0.02 | 3373 | 0.03 | 2656 | 3376 | 1.00 | 1.00 |
| lhr34c | 4264 | 0.05 | 3571 | 0.03 | 4405 | 3668 | 1.03 | 1.02 |
| lhr71c | 9142 | 0.02 | 7189 | 0.02 | 9557 | 7377 | 1.04 | 1.02 |
| mark3jac120sc | 13333 | 0.02 | 12963 | 0.04 | 13386 | 12998 | 1.00 | 1.00 |
| mark3jac140sc | 15380 | 0.02 | 15093 | 0.01 | 15453 | 15136 | 1.00 | 1.00 |
| bayer01 | 1253 | 0.03 | 2220 | 0.04 | 1253 | 2220 | 1.00 | 1.00 |
| sinc18 | 26505 | 0.05 | 31722 | 0.04 | 27427 | 31926 | 1.03 | 1.00 |
| sinc15 | 12596 | 0.03 | 15367 | 0.04 | 12917 | 15455 | 1.02 | 1.00 |
| Zhao2 | 12258 | 0.01 | 14069 | 0.02 | 12588 | 14434 | 1.02 | 1.02 |
| mult_dcop_03 | 713 | 0.01 | 940 | 0.07 | 714 | 906 | 1.00 | 0.96 |
| twotone | 7552 | 0.01 | 8458 | 0.02 | 7552 | 8458 | 1.00 | 1.00 |
| onetone1 | 2913 | 0.02 | 3204 | 0.02 | 2921 | 3204 | 1.00 | 1.00 |
| torso1 | 30656 | 0.02 | 34200 | 0.01 | 30656 | 34326 | 1.00 | 1.00 |
| bbmat | 38088 | 0.10 | 46436 | 0.14 | 38888 | 46471 | 1.02 | 1.00 |
| shermanACb | 362 | 0.01 | 426 | 0.01 | 362 | 426 | 1.00 | 1.00 |
| Mean/Median | 0.88/0.87 | | | | 0.89/0.87 | | | |

- Thanks to the one-way variable elimination, `CMLS` can eliminate all the elements in both the strongly reducible and the weakly reducible situations. This is well illustrated by the mult_dcop_3 matrix, which has 7448 irreducible components. `DMLS` and `CMLS` detect 875 singletons during a common preprocessing step. Then during ordering, `DMLS` detects 95 additional blocks versus 229 blocks for `CMLS`.
- `CMLS` can create a row (column) supervariable if two rows (columns) have the same structure. `DMLS` can create a supervariable only if both rows and columns have the same structure. Thus, on the same quotient graph `CMLS` will detect more supervariables than `DMLS`. Note that the use of supervariables improves the accuracy of the structural metric. For example, if we consider that variables $i$ and $j$ belong to the same row supervariable, then the entries in $\mathcal{A}_{i*}$ and $\mathcal{A}_{j*}$ will not be counted as fill-in.

We should stress that these algorithmic differences were justified because `CMLS` is designed to handle more general and complex situations than `DMLS`. What was not at all predicted is that even a `DMLS`-like algorithm—pivot choice limited to the diagonal—could benefit from the more general framework of the `CMLS` ordering.

### 5.3. Structural strategy.

**5.3.1. Structure of the factors.** In this section, we analyze the effect of the ordering on the size of the factors and compare the predicted size and the actual size of the factors. When there are no off-diagonal pivoting and node amalgamation, the actual size would be the same as the predicted size.

Table 5.2 compares `CMLS` with `DMLS` for both the estimated and the real size of the factors, using the `MA41_UNS` solver. For most matrices, the `CMLS` ordering results in sparser factors. The gains in sparsity vary from −22% to 56%, with gains very much comparable for both the estimated and the real size of the factors. On all matrices `CMLS` is either comparable or significantly better than `DMLS` except for lhr34c and lhr71c for which `DMLS` performs better. For the two matrices `CMLS` performs many mass eliminations (approximatively 25% of the variables are eliminated during mass elimination). Instead of dividing our minimum fill-in estimation by the minimum of the sizes of the column and row supervariables, one could anticipate the number of mass eliminations and divide the `AMFI` estimation by the maximum of the sizes of the column and row supervariables. With this modification, we observed similar results in terms of fill-in between `CMLS` and `DMLS` for these two matrices.

As expected, the fact that more flexibility has been offered to select off-diagonal pivots in the constraint matrix helps `CMLS` to preserve the sparsity of the factors. However, in doing so we have allowed `CMLS` to select pivots that do not belong to the maximum weighted matching. Since a structural metric is then used by `CMLS` to select pivots, it is thus critical to evaluate the numerical quality of this pivot sequence with `MA41_UNS`. We recall that, thanks to partial threshold pivoting, the factorization phase of `MA41_UNS` (the default value of the threshold is used in all experiments) will modify the pivot sequence to control the growth of the size of the factors. This may result in an increase in the estimated factor size and number of operations. We thus also provide in Table 5.2 the ratio between the number of nonzeros in the factors and the forecast number of nonzeros in the factors. Note that from a software point of view it is also critical for the estimation to reflect reality. Clearly an accurate estimation is important for algorithms that are implemented without dynamic memory allocation. Even for C, C++, or Fortran 90 based implementations that allow dynamic memory allocations, their cost may be not negligible. Finally, the accuracy of the memory estimation is even more critical in a distributed memory environment. For example, the buffers used for communications need to be well estimated. We see in Table 5.2 that the increase in the size of the factors is reasonable.

**5.3.2. Run-time and memory usage.** In this section, we examine the number of operations, run-time, and memory usage of `MA41_UNS`. The extra cost due to numerical pivoting during factorization is always included in the number of operations. Note that the timings for the factorization and the solution phases have to be interpreted carefully because they strongly depend on the basic linear algebra kernels used.

We see in Table 5.3 that on almost all the matrices, the `CMLS` ordering reduces the amount of memory used, with an average reduction around 11%. The reduction in the number of operations is even larger (median value of 20%) and will contribute to the reduction in the factorization time.

Table 5.4 then compares the time of the three main steps of the solution process. Note that the ordering time of both orderings depends on two opposite effects that are difficult to assess. The better we preserve sparsity, the smaller might be the quotient graph, and the faster we can process it. On the other hand, the better we preserve sparsity, the fewer elements are absorbed, the fewer supervariables are detected, and the higher the complexity might be. However, our new ordering is a real unsymmetric ordering that selects off-diagonal pivots and updates a constraint matrix. One should thus expect the time spent in the ordering to be higher with `CMLS` than with `DMLS`. Indeed, `CMLS` performs more metric computations and has to explicitly store and manipulate the constraint matrix $\mathbf{C}$. The metric update is the most costly step of the ordering so that the complexity of the ordering is tightly linked to the size of $\mathbf{C}$. Considering that the size of $\mathbf{C}^0$ is typically between $2n$ and $3n$, we see in Table 5.4 that `CMLS` is quite competitive with respect to `DMLS` (we observe that the cost of `CMLS`

TABLE 5.3

`MA41_UNS` *memory used (in thousands of reals) and number of operations (in millions). Ratio: Ratio of* `CMLS` *statistic /* `DMLS` *statistic. Mean (resp., Median): Mean (resp., median) of the ratio* `CMLS` *statistic /* `DMLS` *statistic.*

| | Memory needed | | | Number of operations | | |
|---|---|---|---|---|---|---|
| Matrix | `CMLS` | `DMLS` | ratio | `CMLS` | `DMLS` | ratio |
| av41092 | 7104 | 9998 | 0.71 | 1760 | 3533 | 0.49 |
| g7jac200sc | 29082 | 32912 | 0.88 | 27717 | 32553 | 0.85 |
| g7jac180sc | 26500 | 27566 | 0.96 | 24272 | 28190 | 0.86 |
| jan99jac120sc | 3245 | 4615 | 0.70 | 1042 | 1691 | 0.61 |
| jan99jac100sc | 2711 | 3579 | 0.69 | 867 | 1196 | 0.72 |
| lhr34c | 4501 | 3684 | 1.22 | 731 | 422 | 1.73 |
| lhr71c | 9786 | 7465 | 1.31 | 1949 | 852 | 2.28 |
| mark3jac120sc | 13909 | 13511 | 1.02 | 7524 | 6465 | 1.16 |
| mark3jac140sc | 15937 | 15963 | 1.00 | 8592 | 7558 | 1.13 |
| bayer01 | 1256 | 2231 | 0.56 | 41 | 140 | 0.29 |
| sinc18 | 32375 | 35409 | 0.91 | 49281 | 60152 | 0.81 |
| sinc15 | 15031 | 16885 | 0.93 | 16515 | 19376 | 0.85 |
| Zhao2 | 13200 | 15492 | 0.85 | 7622 | 9655 | 0.78 |
| mult_dcop_03 | 746 | 969 | 0.76 | 51 | 117 | 0.43 |
| twotone | 8038 | 9006 | 0.89 | 4791 | 5412 | 0.88 |
| onetone1 | 3437 | 3652 | 0.94 | 1035 | 1284 | 0.80 |
| torso1 | 33759 | 36761 | 0.91 | 24620 | 36523 | 0.67 |
| bbmat | 39303 | 47156 | 0.83 | 31576 | 54061 | 0.58 |
| shermanACb | 394 | 452 | 0.87 | 20 | 30 | 0.66 |
| Mean/Median | | | 0.89/0.89 | | | 0.87/0.80 |

TABLE 5.4

`MA41_UNS` *ordering, factorization and solution time (in seconds). Ratio: Ratio of* `CMLS` *statistic /* `DMLS` *statistic. Mean (resp., Median): Mean (resp., median) of the ratio* `CMLS` *statistic /* `DMLS` *statistic.*

| | Ordering time | | | Factorization time | | | Solution time | | |
|---|---|---|---|---|---|---|---|---|---|
| Matrix | `CMLS` | `DMLS` | ratio | `CMLS` | `DMLS` | ratio | `CMLS` | `DMLS` | ratio |
| av41092 | 3.38 | 2.72 | 1.24 | 1.95 | 2.98 | 0.65 | 0.042 | 0.051 | 0.82 |
| g7jac200sc | 27.13 | 9.96 | 2.72 | 23.37 | 25.63 | 0.91 | 0.125 | 0.135 | 0.92 |
| g7jac180sc | 24.34 | 8.11 | 3.00 | 22.15 | 24.04 | 0.92 | 0.113 | 0.119 | 0.94 |
| jan99jac120sc | 5.29 | 3.01 | 1.75 | 1.48 | 2.04 | 0.72 | 0.037 | 0.043 | 0.86 |
| jan99jac100sc | 4.16 | 2.03 | 2.04 | 1.14 | 1.83 | 0.62 | 0.028 | 0.031 | 0.90 |
| lhr34c | 5.01 | 2.27 | 2.20 | 1.22 | 0.95 | 1.28 | 0.041 | 0.037 | 1.10 |
| lhr71c | 11.19 | 5.13 | 2.18 | 3.16 | 2.33 | 1.35 | 0.096 | 0.084 | 1.14 |
| mark3jac120sc | 8.26 | 3.59 | 2.30 | 5.88 | 4.90 | 1.20 | 0.069 | 0.070 | 0.98 |
| mark3jac140sc | 9.84 | 4.18 | 2.35 | 6.77 | 5.85 | 1.15 | 0.083 | 0.081 | 1.02 |
| bayer01 | 1.66 | 1.19 | 1.39 | 0.36 | 0.49 | 0.73 | 0.037 | 0.039 | 0.94 |
| sinc18 | 25.63 | 12.66 | 2.02 | 33.72 | 32.10 | 1.05 | 0.079 | 0.077 | 1.02 |
| sinc15 | 10.89 | 4.60 | 2.36 | 10.98 | 10.79 | 1.01 | 0.040 | 0.040 | 1.00 |
| Zhao2 | 2.22 | 0.93 | 2.38 | 5.67 | 6.97 | 0.81 | 0.052 | 0.053 | 0.98 |
| mult_dcop_03 | 0.86 | 0.42 | 2.04 | 0.22 | 0.25 | 0.88 | 0.014 | 0.013 | 1.07 |
| twotone | 3.59 | 2.28 | 1.57 | 6.05 | 7.06 | 0.85 | 0.094 | 0.109 | 0.86 |
| onetone1 | 1.52 | 0.50 | 3.04 | 1.25 | 1.70 | 0.73 | 0.026 | 0.030 | 0.86 |
| torso1 | 14.49 | 70.20 | 0.20 | 15.87 | 25.25 | 0.62 | 0.180 | 0.191 | 0.94 |
| bbmat | 40.88 | 14.07 | 2.90 | 41.42 | 48.74 | 0.84 | 0.190 | 0.184 | 1.03 |
| shermanACb | 0.31 | 0.14 | 2.21 | 0.10 | 0.11 | 0.90 | 0.009 | 0.009 | 1.00 |
| Mean/Median | | | 2.1/2.2 | | | 0.91/0.88 | | | 0.97/0.98 |

does not linearly increase with the size of $\mathbf{C}^0$). Two algorithmic differences might explain the good behavior of the `CMLS` ordering (see, for example, torso1 matrix).

- Since `CMLS` has the flexibility to select pivots in the constraint matrix it may not be critical to know the metric of the entries that belong to a fairly dense row or column. That is why our CMLS implementation can easily avoid metric updates of such entries in the **C** matrix.

- Furthermore, supervariables have been generalized in our context resulting in separated row and column supervariables. This feature helps CMLS exploit the unsymmetric structure of the matrix in a more efficient way.

We then see in Table 5.3 that the decrease in fill-in and in the number of operations performed during the factorization phase leads to a decrease in the factorization time. The reductions in factorization time are slightly smaller than those in the number of operations (the average reduction in the number of operations is around 20%). This is because sparser factors often lead to smaller full blocks for which basic linear algebra kernels are slower. We observed that the flop rate of MA41_UNS tends to be smaller with CMLS than with DMLS: the average flop rate is nearly 1.02 GFlops with the DMLS ordering, whereas it is around 0.99 GFlops with the CMLS ordering.

### 5.4. Impact of the hybrid strategies on SuperLU_DIST.

We now study the numerical behavior of SuperLU_DIST using CMLS ordering. Because of the static pivoting strategy used during factorization, SuperLU_DIST is expected to be numerically more sensitive than MA41_UNS to the use of hybrid strategies in pivot selection, and iterative refinement may be required to obtain an accurate solution, as was observed in [4]. We thus analyze the componentwise backward error of the solution [8] during iterative refinement. Note that one step of iterative refinement costs at least as much as one forward and backward substitution. The cost of the solution phase is closely related to the number of steps of iterative refinement. In the hybrid strategy (see section 4.1), a relative threshold is set to avoid the selection of small pivots in **C**. This was chosen to be 0.01 in all our experiments.

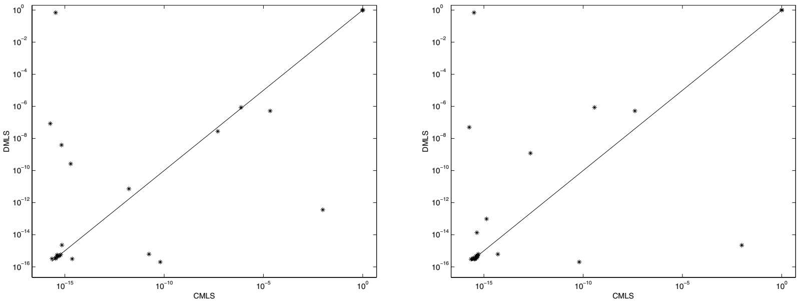
Table 5.5 shows that SuperLU_DIST often does not compute an accurate solution if the CMLS ordering is obtained with a structural metric (compare the number of entries with $\star$ in columns STR and HYB). With the hybrid strategy to select pivots, some more pivots are postponed by CMLS because of their numerical values. We observe that this often results in an increase in the fill-in in the factor with respect to a structural metric (compare columns STR and HYB) but improves the numerical reliability of the CMLS pivot sequence. Note that we report only the median values because large variations of gains perturb the average statistics.

Figure 5.1 compares the componentwise backward errors during iterative refinement with the CMLS and DMLS orderings (results after two and four steps). In each plot, a data point above the diagonal corresponds to a matrix for which CMLS performs better in terms of componentwise backward error. Is it clear that using the CMLS ordering improves the numerical behavior of SuperLU_DIST. There are still two matrices (av41092 and Zhao2) for which, with either CMLS or DMLS ordering, iterative refinement does not converge to an accurate solution (upper right corner). The torso1 matrix is the only one for which the DMLS approach succeeds, whereas the CMLS approach fails (bottom right corner). There are four matrices in the upper left corner (lhr34c, lhr71c, mult_dcop3, and fidapm11) for which the backward error of SuperLU_DIST combined with DMLS remains larger than $10^{-8}$, whereas SuperLU_DIST combined with CMLS converges in less than four iterations. It is interesting to observe that the only matrices for which CMLS with the hybrid strategy leads to significantly more fill-in in the factors are the lhr34c, lhr71c, and mult_dcop3 matrices on which DMLS did not converge after iterative refinement (and independently of the number of steps). Note finally that, with CMLS, on all problems except three we obtain an accurate solution (backward error smaller than $10^{-8}$) with four steps of iterative refinement, whereas,

Table 5.5

`SuperLU_DIST` *size of the factors (in thousands) and number of operations (in millions).* `STR`*: The pivots are selected according to the* `AMFI` *structural metric.* `HYB`*: The pivots are selected using a hybrid strategy. Ratio: Ratio of* `CMLS` *statistic /* `DMLS` *statistic. Median: Median value of the ratio* `CMLS` *statistic /* `DMLS` *statistic. ⋆: After iterative refinement, the backward error is greater than* $10^{-8}$.

| | Size of factors | | | | Number of operations | | | |
|---|---|---|---|---|---|---|---|---|
| Matrix | `STR` | ratio | `HYB` | ratio | `STR` | ratio | `HYB` | ratio |
| av41092 | ⋆5773 | 0.72 | ⋆5993 | 0.74 | 1.12e+09 | 0.46 | 1.28e+09 | 0.52 |
| g7jac200sc | ⋆20394 | 0.92 | 20404 | 0.93 | 1.08e+10 | 1.18 | 1.11e+10 | 1.21 |
| g7jac180sc | ⋆19537 | 1.01 | 17426 | 0.90 | 1.26e+10 | 1.47 | 8.75e+09 | 1.02 |
| jan99jac120sc | 1923 | 0.82 | 1923 | 0.82 | 2.57e+08 | 0.64 | 2.57e+08 | 0.64 |
| jan99jac100sc | 1532 | 0.87 | 1532 | 0.87 | 2.04e+08 | 0.80 | 2.04e+08 | 0.80 |
| lhr34c | ⋆3342 | 1.19 | 3645 | 1.30 | 2.80e+08 | 2.09 | 4.19e+08 | 3.13 |
| lhr71c | ⋆7241 | 1.15 | 9434 | 1.50 | 7.25e+08 | 1.73 | 2.07e+09 | 4.93 |
| mark3jac120sc | 10847 | 1.04 | 10624 | 1.02 | 4.84e+09 | 1.17 | 4.29e+09 | 1.04 |
| mark3jac140sc | ⋆12552 | 1.01 | 12673 | 1.02 | 5.56e+09 | 1.13 | 5.53e+09 | 1.12 |
| bayer01 | 909 | 0.66 | 909 | 0.66 | 1.80e+07 | 0.37 | 1.80e+07 | 0.37 |
| sinc18 | ⋆24234 | 0.84 | 26702 | 0.93 | 3.46e+10 | 0.82 | 4.18e+10 | 1.00 |
| sinc15 | ⋆11768 | 0.88 | 13447 | 1.00 | 1.17e+10 | 0.86 | 1.50e+10 | 1.10 |
| Zhao2 | ⋆10954 | 0.87 | ⋆11174 | 0.89 | 5.94e+09 | 0.78 | 6.36e+09 | 0.84 |
| mult_dcop_03 | 524 | 1.40 | 520 | 1.39 | 1.45e+07 | 4.84 | 1.30e+07 | 4.35 |
| twotone | 7118 | 0.90 | 7030 | 0.89 | 4.45e+09 | 1.06 | 4.38e+09 | 1.05 |
| onetone1 | 2579 | 0.93 | 3020 | 1.09 | 7.98e+08 | 0.83 | 1.05e+09 | 1.10 |
| torso1 | ⋆30156 | 0.88 | ⋆29455 | 0.86 | 2.35e+10 | 1.05 | 1.98e+10 | 0.88 |
| fidapm11 | ⋆20777 | 0.99 | 21373 | 1.02 | 1.31e+10 | 0.96 | 1.41e+10 | 1.04 |
| bbmat | 36522 | 0.70 | 43074 | 0.82 | 2.07e+10 | 0.33 | 2.92e+10 | 0.47 |
| shermanACb | 342 | 0.85 | 347 | 0.87 | 1.62e+07 | 0.64 | 1.74e+07 | 0.69 |
| cavity16 | ⋆321 | 0.77 | 330 | 0.79 | 1.86e+07 | 0.57 | 1.91e+07 | 0.58 |
| INV-EXTRUSION-1 | ⋆25550 | 1.06 | 25973 | 1.08 | 2.15e+10 | 1.19 | 2.26e+10 | 1.25 |
| MIXING-TANK | ⋆44762 | 1.07 | 41760 | 1.00 | 7.54e+10 | 1.18 | 6.50e+10 | 1.02 |
| Median | | 0.89 | | 0.92 | | 0.96 | | 1.01 |



(a) Component-wise backward, step 2. (b) Component-wise backward, step 4.

Fig. 5.1. `SuperLU_DIST` *componentwise backward error during iterative refinement.*

with `DMLS`, iterative refinement did not converge on six matrices.

**6. Concluding remarks.** The originality of the `CMLS` algorithm relies on its ability to compute an unsymmetric permutation with the following goals in mind: to reduce the fill-in in the factors and to preselect numerically good pivots for the factorization. It is based on a constraint matrix which contains the candidate pivots and a quotient graph that is used to compute the structural metrics. The `CMLS` algorithm can be used to design a family of orderings that can address a large class of problems. The main results and the properties of the algorithm are summarized as follows:

- Significant reductions in terms of fill-in (13%) and flops (20%) have been obtained with the structural strategy to select the pivots.
- Using structural metrics to select the pivots does not affect the numerical behavior of the `MA41_UNS` solver.
- On numerically difficult problems, `CMLS` can be used to improve the accuracy of `SuperLU_DIST` and reduce the number of steps of iterative refinement during the solution phase.
- Our generalized supervariables could be used in the context of `DMLS` to also improve the metric computation.

One indirect but important consequence of our work is that we do not need to limit our pivot choice to a maximum weighted transversal of the original matrix. Preliminary experiments have shown that the maximum weighted matching can in fact be replaced by a simpler structural maximum transversal during the preprocessing phase (Step 1 as defined in section 2). One possible direction for future work could then be to design a parallel version of the preprocessing phase.

Furthermore, the constraint matrix $\mathbf{C}$ contains the information of an incomplete factorization. We intend to use it as a preconditioner and to compare its quality and cost with existing incomplete $\mathbf{LU}$ factorizations.

Finally, in our paper we have focused on local strategies and on very unsymmetric matrices. We also did experiments to compare our algorithms with global strategies such as nested dissection and observed that our approach was generally better on this set of test matrices. Combining our numerically based local heuristics with structurally based global strategies is another interesting direction for future work.

**Acknowledgments.** We want to thank Cleve Ashcraft and the two anonymous referees for their useful and constructive indications on how to improve the presentation of this paper.

## REFERENCES

[1] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 886–905.

[2] P. R. Amestoy and I. S. Duff, *Vectorization of a multiprocessor multifrontal code*, International Journal of Supercomputer Applications, 3 (1989), pp. 41–59.

[3] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.

[4] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and X. S. Li, *Analysis and comparison of two general sparse solvers for distributed memory computers*, ACM Trans. Math. Software, 27 (2001), pp. 388–421.

[5] P. R. Amestoy, X. S. Li, and E. Ng, *Diagonal Markowitz scheme with local symmetrization*, Tech. rep. RT/APO/03/5, ENSEEIHT-IRIT, Toulouse, France, 2003. Also appeared as Lawrence Berkeley Lab report LBNL-53854, Berkeley, CA, 2003.

[6] P. R. Amestoy, X. S. Li, and S. Pralet, *Constrained Markowitz with local symmetrization*, Technical rep. RT/APO/04/05, ENSEEIHT-IRIT, Toulouse, France, 2004. Also appeared as Lawrence Berkeley Lab report LBNL-56861, Berkeley, CA, 2005, and CERFACS report TR/PA/04/137, Toulouse, France, 2004.

[7] P. R. Amestoy and C. Puglisi, *An unsymmetrized multifrontal LU factorization*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 553–569.

[8] M. Arioli, J. Demmel, and I. S. Duff, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.

[9] C. Ashcraft, *Compressed graphs and the minimum degree algorithm*, SIAM J. Sci. Comput., 16 (1995), pp. 1404–1411.

[10] C. Ashcraft and R. G. Grimes, *SPOOLES: An object-oriented sparse matrix library*, in Proceedings of the Ninth Annual SIAM Conference on Parallel Processing for Scientific Computing (San Antonio, TX), SIAM, Philadelphia, 1999.

[11] T. A. Davis, *University of Florida sparse matrix collection*, http://www.cise.ufl.edu/research/sparse/matrices, 2002.

[12] T. A. Davis, *Algorithm* 832*: UMFPACK V*4.3*—an unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 196–199.

[13] T. A. Davis and I. S. Duff, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 140–158.

[14] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[15] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Oxford University Press, London, 1986.

[16] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.

[17] I. S. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.

[18] I. S. Duff and J. K. Reid, *A comparison of sparsity orderings for obtaining a pivotal sequence in Gaussian elimination*, J. Inst. Math. Appl., 14 (1974), pp. 281–291.

[19] I. S. Duff and J. K. Reid, *MA*27*—a set of Fortran subroutines for solving sparse symmetric sets of linear equations*, Technical rep. R.10533, AERE, Harwell, England, 1982.

[20] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[21] I. S. Duff and J. K. Reid, *The multifrontal solution of unsymmetric sets of linear equations*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 633–641.

[22] I. S. Duff and J. K. Reid, *MA*47*, a Fortran code for direct solution of indefinite sparse symmetric linear systems*, Tech. Rep. RAL 95-001, Rutherford Appleton Laboratory, Chilton, Didcot, Oxon, UK, 1995.

[23] A. George and J. W. H. Liu, *A fast implementation of the minimum degree algorithm using quotient graphs*, ACM Trans. Math. Software, 6 (1980), pp. 337–358.

[24] A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1981.

[25] A. George and D. R. McIntyre, *On the application of the minimum degree algorithm to finite element systems*, SIAM J. Numer. Anal., 15 (1978), pp. 90–112.

[26] A. Gupta, *Recent advances in direct methods for solving unsymmetric sparse systems of linear equations*, ACM Trans. Math. Software, 28 (2002), pp. 301–324.

[27] X. S. Li and J. W. Demmel, *A scalable sparse direct solver using static pivoting*, in Proceedings of the Ninth Annual SIAM Conference on Parallel Processing for Scientific Computing (San Antonio, TX), SIAM, Philadelphia, 1999.

[28] X. S. Li and J. W. Demmel, *SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Software, 29 (2003), pp. 110–140.

[29] J. W. H. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. Math. Software, 11 (1985), pp. 141–153.

[30] E. Ng and P. Raghavan, *Performance of greedy ordering heuristics for sparse Cholesky factorization*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 902–914.

[31] G. Pagallo and C. Maulino, *A bipartite quotient graph model for unsymmetric matrices*, in Numerical Methods, Lecture Notes in Math. 1005, Springer-Verlag, Berlin, 1983, pp. 227–239.

[32] S. Pralet, *Constrained Orderings and Scheduling for Parallel Sparse Linear Algebra*, Ph.D. thesis, Institut National Polytechnique de Toulouse, Toulouse, France, 2004. Available as CERFACS technical report, TH/PA/04/105, 2004.

[33] E. Rothberg and S. C. Eisenstat, *Node selection strategies for bottom-up sparse matrix ordering*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 682–695.

[34] A. F. van der Stappen, R. H. Bisseling, and J. G. G. van de Vorst, *Parallel sparse LU decomposition on a mesh network of transputers*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 853–879.

[35] R. C. Whaley and A. Petitet, *Minimizing development and maintenance costs in supporting persistently optimized BLAS*, Software: Practice and Experience, 35 (2005), pp. 101–121. Also available online from http://www.cs.utsa.edu/~whaley/papers/spercw04.ps.

[36] R. C. Whaley, A. Petitet, and J. J. Dongarra, *Automated empirical optimization of software and the ATLAS project*, Parallel Computing, 27 (2001), pp. 3–35. Also available as University of Tennessee LAPACK Working Note 147, UT-CS-00-448, University of Tennessee, Knoxville, TN, 2000, http://www.netlib.org/lapack/lawns/lawn147.ps, 2000.

[37] Z. Zlatev, *On some pivotal strategies in Gaussian elimination by sparse technique*, SIAM J. Numer. Anal., 17 (1980), pp. 18–30.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 1, pp. 328–347

# GEOMETRIC MEANS IN A NOVEL VECTOR SPACE STRUCTURE ON SYMMETRIC POSITIVE-DEFINITE MATRICES*

VINCENT ARSIGNY†, PIERRE FILLARD†, XAVIER PENNEC†, AND NICHOLAS AYACHE†

**Abstract.** In this work we present a new generalization of the geometric mean of positive numbers on symmetric positive-definite matrices, called Log-Euclidean. The approach is based on two novel algebraic structures on symmetric positive-definite matrices: first, a lie group structure which is compatible with the usual algebraic properties of this matrix space; second, a new scalar multiplication that smoothly extends the Lie group structure into a vector space structure. From bi-invariant metrics on the Lie group structure, we define the Log-Euclidean mean from a Riemannian point of view. This notion coincides with the usual Euclidean mean associated with the novel vector space structure. Furthermore, this means corresponds to an arithmetic mean in the domain of matrix logarithms. We detail the invariance properties of this novel geometric mean and compare it to the recently introduced affine-invariant mean. The two means have the same determinant and are equal in a number of cases, yet they are not identical in general. Indeed, the Log-Euclidean mean has a larger trace whenever they are not equal. Last but not least, the Log-Euclidean mean is much easier to compute.

**Key words.** geometric mean, symmetric positive-definite matrices, Lie groups, bi-invariant metrics, geodesics

**AMS subject classifications.** 47A64, 26E60, 53C35, 22E99, 32F45, 53C22

**DOI.** 10.1137/050637996

## 1. Introduction.

Symmetric positive-definite (SPD) matrices of real numbers appear in many contexts. In medical imaging, their use has become common during the last 10 years with the growing interest in diffusion tensor magnetic resonance imaging (DT-MRI, or simply DTI) [3]. In this imaging technique, based on nuclear magnetic resonance (NMR), the assumption is made that the random diffusion of water molecules at a given position in a biological tissue is Gaussian. As a consequence, a diffusion tensor image is an SPD matrix-valued image in which the SPD matrix associated with the current volume element (or voxel) is the covariance matrix of the local diffusion process. SPD matrices also provide a powerful framework for modeling the anatomical variability of the brain, as shown in [15]. More generally, they are widely used in image analysis, especially for segmentation, grouping, motion analysis, and texture segmentation [16]. They are also used intensively in mechanics, for example, with strain or stress tensors [4]. Last, but not least, SPD matrices are becoming a common tool in numerical analysis for generating adapted meshes to reduce the computational cost of solving partial differential equations (PDEs) in three dimensions [17].

As a consequence, there has been a growing need to carry out computations with these objects, for instance to interpolate, restore, and enhance images SPD matrices. To this end, one needs to define a complete operational framework. This

---

*Received by the editors August 11, 2005; accepted for publication (in revised form) by L. Reichel August 23, 2006; published electronically February 23, 2007. This work was supported by the INRIA, France.

http://www.siam.org/journals/simax/29-1/63799.html

†ASCLEPIOS Research Project, INRIA, Sophia-Antipolis, FR-06902, France (Vincent.Arsigny@Sophia.inria.fr, Pierre.Fillard@Sophia.inria.fr, Xavier.Pennec@Sophia.inria.fr, Nicholas.Ayache@Sophia.inria.fr).

is necessary to fully generalize to the SPD case the usual statistical tools or PDEs on vector-valued images. The framework of Riemannian geometry [8] is particularly adapted to this task, since many statistical tools [18] and PDEs can be generalized to this framework.

To evaluate the relevance of a given Riemannian metric, the properties of the associated notion of [illegible] are of great importance. Indeed, most computations useful in practice involve averaging procedures. This is the case in particular for the interpolation, regularization, and extrapolation of SPD matrices, where mean values are implicitly computed to generate new data. For instance, the classical regularization technique based on the heat equation is equivalent to the convolution of the original data with Gaussian kernels.

Let $\mathcal{M}$ be an abstract manifold endowed with a Riemannian metric, whose associated distance is $d(.,.)$. Then the classical generalization of the Euclidean mean is given by the [illegible] (also called the [illegible]) [18, 19]. Let $(x_i)_{i=1}^N$ be $N$ points of $\mathcal{M}$. Their Fréchet mean $\mathbb{E}(x_i)$ (possibly not uniquely defined) is defined as the point minimizing the following [illegible]:

$$\mathbb{E}(x_i) = \arg\min_x \sum_{i=1}^{N} d^2(x, x_i). \tag{1.1}$$

One can directly use a Euclidean structure on square matrices to define a metric on the space of SPD matrices. This is straightforward, and in this setting, the Riemannian mean of a system of SPD matrices is their [illegible] mean, which is an SPD matrix since SPD matrices form a convex set. However, this mean is [illegible] adequate in many situations, for two main reasons. First, symmetric matrices with nonpositive eigenvalues are at a finite distance from any SPD matrix in this framework. In the case of DT-MRI, this is not physically acceptable, since this amounts to assuming that small diffusions (i.e., small eigenvalues) are much more likely than large diffusions (i.e., large eigenvalues). A priori, large and small diffusions are equally unlikely in DT-MRI, and a [illegible] should be respected. In particular, a matrix and its inverse should be at the same distance from the identity. Therefore, the use of a generalization to SPD matrices of the [illegible] mean of positive numbers would be preferable, since such a mean is precisely invariant with respect to inversion.

Second, an SPD matrix corresponds typically to a [illegible]. The value of its determinant is a direct measure of the dispersion of the associated multivariate Gaussian. The reason is that the volumes of associated trust regions are proportional to the square root of this determinant. But the Euclidean averaging of SPD matrices often leads to a [illegible]: the determinant of the Euclidean mean can be strictly larger than the original determinants. The reason is that the induced interpolation of determinants is polynomial and [illegible] monotonic in general. In DTI, diffusion tensors are assumed to be covariance matrices of the local Brownian motion of water molecules. Introducing more dispersion in computations amounts to introducing more diffusion, which is physically unacceptable. For illustrations of this effect, see [20, 21]. As a consequence, the determinant of a mean of SPD matrices should remain bounded by the values of the determinants of the averaged matrices.

To fully circumvent these difficulties, other metrics have been recently proposed for SPD matrices. With the affine-invariant metrics proposed in [12, 22, 23, 19], symmetric matrices with negative and null eigenvalues are at an infinite distance from any SPD matrix. The swelling effect has disappeared, and the symmetry with respect to inversion is respected. These new metrics provide an affine-invariant generalization of the geometric mean of positive numbers on SPD matrices. But the price paid for this success is a high computational burden in practice, essentially due to the curvature induced on the space of SPD matrices. This leads in many cases to slow and hard-to-implement algorithms (especially for PDEs) [12].

We propose here a new Riemannian framework on SPD matrices, which gives rise to a novel generalization of the geometric mean to SPD matrices. It fully overcomes the computational limitations of the affine-invariant framework, while conserving excellent theoretical properties. This is obtained with a new family of metrics named [illegible]. Such metrics are particularly simple to use. They result in classical Euclidean computations in the domain of matrix logarithms. As a consequence, there is a closed form for the Log-Euclidean mean, contrary to the affine-invariant case. It results in a drastic reduction in computation time: the Log-Euclidean mean can be computed approximately 20 times faster.

The remainder of this article is organized as follows. In section 2, we recall a number of elementary properties of the space of SPD matrices. Then we proceed in section 3 to the theory of Log-Euclidean metrics which is based on two novel algebraic structures on SPD matrices: a Lie group structure and a new scalar multiplication which complements the new multiplication to obtain a new vector space structure. The definition of the Log-Euclidean mean is deduced from these new structures. Contrary to the affine-invariant mean, there is a closed form for the Log-Euclidean mean and it is simple to compute. In section 4 we highlight the resemblances and differences between affine-invariant and Log-Euclidean means. They are quite similar, since they have the same determinant, which is the classical geometric mean of the determinants of the averaged SPD matrices. They even coincide in a number of cases, and yet are different in general. We prove that Log-Euclidean means are strictly more anisotropic when averaged SPD matrices are isotropic enough.

## 2. Preliminaries.

We begin with a description of the fundamental properties and tools used in this work. First, we recall the elementary properties of the matrix exponential. Then we examine the general properties of SPD matrices. These properties are of two types: algebraic and differential. On the one hand, SPD matrices have algebraic properties because they are a special kind of invertible matrices, and on the other hand they can be considered globally as a smooth manifold and therefore have differential geometry properties. These properties are not independent: on the contrary, they are compatible in a profound way. This compatibility is the core of the approach developed here.

### 2.1. Notation.

We will use the following definitions and notation:

- $Sym_\star^+(n)$ is the space of SPD real $n \times n$ matrices.
- $Sym(n)$ is the vector space of real $n \times n$ symmetric matrices.
- $GL(n)$ is the group of real invertible $n \times n$ matrices.
- $M(n)$ is the space of real $n \times n$ square matrices.
- $\mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ is the diagonal matrix constructed with the real values $(\lambda_i)_{i \in 1 \ldots n}$ in its diagonal.
- For any square matrix $M$, $Sp(M)$ is the [illegible] of $M$, i.e., the set of its eigenvalues.
- $\phi : E \rightarrow F$ is differentiable mapping between two smooth manifolds. Its differential at a point $M \in E$ acting on a infinitesimal displacement $dM$ in the tangent space to $E$ at $M$ is written as $D_M \phi . dM$.

**2.2. Matrix exponential.** The exponential plays a central role in [illegible] (see [11, 5, 8]). We will consider here only the matrix version of the exponential, which is a tool that we extensively use in the next sections. We recall its definition and give its elementary properties. Last but not least, we give the Baker–Campbell–Hausdorff formula. It is a powerful tool that provides fine information on the structure of Lie groups around the identity. We will see in section 4 how it can be used to compare Log-Euclidean means to affine-invariant means in terms of anisotropy.

DEFINITION 2.1. [illegible] $\exp(M)$ [illegible] $M$ [illegible] $\exp(M) = \sum_{n=0}^{\infty} \frac{M^k}{k!}$ [illegible] $G \in GL(n)$ [illegible] $M \in M(n)$ [illegible] $G = \exp(M)$ [illegible] $M$ [illegible] $N$

In general, the logarithm of a real invertible matrix may not exist, and if it exists it may not be unique. The lack of existence is a general phenomenon in connected Lie groups. One generally needs [illegible] exponentials to reach every element [10]. The lack of uniqueness is essentially due to the influence of rotations: rotating of an angle $\alpha$ is the same as rotating of an angle $\alpha + 2k\pi$, where $k$ is an integer. Since the logarithm of a rotation matrix directly depends on its rotation angles (one angle suffices in three dimensions, but several angles are necessary when $n > 3$), it is not unique. However, when a real invertible matrix has no (complex) eigenvalue on the (closed) negative real line, then it has a unique real logarithm whose (complex) eigenvalues have an imaginary part in $]-\pi, \pi[$ [2]. This particular logarithm is called [illegible]. We will write $\log(M)$ for the principal logarithm of a matrix $M$ whenever it is defined.

THEOREM 2.2. $\exp : M(n) \rightarrow GL(n)$ [illegible] $\mathcal{C}^{\infty}$ [illegible] $M \in M(n)$ [illegible] $dM \in M(n)$ [illegible]

$$D_M \exp .dM = \sum_{k=1}^{\infty} \frac{1}{k!} \left( \sum_{l=0}^{k-1} M^{k-l-1}.dM.M^l \right). \tag{2.1}$$

[illegible]. The smoothness of exp is simply a consequence of the uniform absolute convergence of its series expansion in any compact set of $M(n)$. The differential is obtained classically by a term by term derivation of the series defining the exponential. □

We see here that the noncommutativity of the matrix multiplication seriously complicates the differentiation of the exponential, which is much simpler in the scalar case. However, taking the trace in (2.1) yields the following.

COROLLARY 2.3. [illegible]

$$\text{Trace}(D_M \exp .dM) = \text{Trace}(\exp(M).dM). \tag{2.2}$$

In the following we will also use this property on determinants.

PROPOSITION 2.4. [illegible] $M \in M(n)$ [illegible] $\det(\exp(M)) = \exp(\text{Trace}(M))$

[illegible]. This is easily seen in terms of eigenvalues of $M$. The Jordan decomposition of $M$ [1] ensures that Trace($M$) is the sum of its eigenvalues. But the exponential of a triangular matrix transforms the diagonal values of this matrix into their scalar exponential. The determinant of $\exp(M)$ is simply the product of its eigenvalues, which is precisely the exponential of the trace of $M$. □

THEOREM 2.5 (Baker–Campbell–Hausdorff formula [9] (matrix case)). [illegible] $M, N \in M(n)$ [illegible] $t \in \mathbb{R}$ [illegible] $t$ [illegible]

[illegible]

$$\begin{aligned}\log(\exp(t.M).\exp(t.N)) &= t.(M+N) + t^2/2([M,N]) \\ &\quad + t^3/12([M,[M,N]] + [N,[N,M]]) \\ &\quad + t^4/24([[M,[M,N]],N]) + O(t^5). \end{aligned} \tag{2.3}$$

[illegible] $[M,N] = MN - NM$ [illegible] $M$ [illegible] $N$

The Baker–Campbell–Hausdorff formula shows how much $\exp(\log(M).\log(N))$ deviates from $M+N$ due to the noncommutativity of the matrix product. Remarkably, this deviation can be expressed only in terms of Lie brackets between $M$ and $N$ [14].

**2.3. Algebraic properties.** SPD matrices have remarkable algebraic properties. First, there always exists a unique real and [illegible] logarithm for any SPD matrix, which is its principal logarithm. Second, if the space of SPD matrices is not a subgroup of $GL(n)$, it is stable with respect to inversion. Moreover, its spectral decomposition is particularly simple.[1]

THEOREM 2.6. [illegible] $S \in Sym(n)$ [illegible] $S$ [illegible] $Sym_\star^+(n)$ [illegible] $GL(n)$ [illegible] $\exp : Sym(n) \rightarrow Sym_\star^+(n)$ [illegible]

[illegible]. For a proof of the first assertion, see elementary linear algebra manuals, or [1]. For the second assertion, we see from section 2.2 that SPD matrices have a unique real logarithm whose eigenvalues have an imaginary part between $-\pi$ and $+\pi$, since the eigenvalues of SPD matrices are real and always positive. The principal logarithm of an SPD matrix can be obtained simply by replacing its eigenvalues with their natural logarithms, which shows that this logarithm is symmetric. $\square$

Thanks to the existence of an orthonormal basis in which an SPD matrix (resp., a symmetric matrix) is diagonal, the logarithm (resp., the exponential) has a particularly simple expression. In such a basis, taking the log (resp., the exp) is simply done by applying its scalar version to eigenvalues:

$$\begin{cases} \log(R.\mathrm{Diag}(\lambda_1,\ldots,\lambda_N).R^T) = R.\mathrm{Diag}(\log(\lambda_1),\ldots,\log(\lambda_N)).R^T, \\ \exp(R.\mathrm{Diag}(\lambda_1,\ldots,\lambda_N).R^T) = R.\mathrm{Diag}(\exp(\lambda_1),\ldots,\exp(\lambda_N)).R^T. \end{cases}$$

These formulae provide a particularly efficient method to calculate the logarithms and exponentials of symmetric matrices, whenever the cost of a diagonalization is less than that of the many matrix multiplications (in the case of the exponential) and inversions (in the case of the logarithm) used in the general matrix case by classical algorithms [13, 24]. For small values of $n$, and in particular $n = 3$, we found such formulae to be extremely useful.

**2.4. Differential properties.** From the point of view of topology and differential geometry, the space of SPD matrices also has many particularities. The properties recalled here are elementary and will not be detailed. See [25] for complete proofs.

PROPOSITION 2.7. $Sym_\star^+(n)$ [illegible] $Sym(n)$ [illegible] $Sym(n)$ [illegible] $n(n+1)/2$

**2.5. Compatibility between algebraic and differential properties.** We have seen that exp is a smooth bijection. We show here that the logarithm, i.e.,

---

[1] This is due to the fact that SPD matrices are *normal operators*, like rotations and antisymmetric matrices [1].

its inverse, is also smooth. As a consequence, all the algebraic operations on SPD matrices presented before are also smooth, in particular the inversion. Thus, the two structures are fully compatible.

THEOREM 2.8. $\log : Sym^+_\star(n) \to Sym(n)$ [illegible] $\mathcal{C}^\infty$ [illegible] exp [illegible] log [illegible] exp [illegible]

[illegible]. In fact, we need only prove the last assertion. If it is true, the implicit function theorem [6] applies and ensures that log is also smooth. Since the differential of exp at 0 is simply given by the identity, it is invertible by continuity in a neighborhood of 0. We now show that this propagates to the entire space $Sym(n)$. Indeed, let us then suppose that for a point $M$, the differential $D_{M/2}\exp$ is invertible. We claim that then $D_M \exp$ is also invertible, which suffices to prove the point. To show this, let us take $dM \in Sym(n)$ such that $D_M \exp .dM = 0$. If $D_M \exp$ is invertible, we should have $dM = 0$. To see this, remark that $\exp(M) = \exp(M/2).\exp(M/2)$. By differentiation and applying to $dM$, we get

$$D_M \exp .dM = 1/2((D_{M/2}\exp .dM).\exp(M/2) + \exp(M/2).(D_{M/2}\exp .dM)) = 0.$$

This implies by multiplication by $\exp(-M/2)$:

$$\exp(-M/2)(D_{M/2}\exp .dM).\exp(M/2) + (D_{M/2}\exp .dM) = 0.$$

Since

$$A^{-1}.\exp(B).A = \exp(A^{-1}.B.A)$$

we have also by differentiation

$$A^{-1}.D_B \exp(dB).A = D_B \exp(A^{-1}.dB.A).$$

Using this simplification and the hypothesis that $D_{M/2}\exp$ is invertible, we obtain

$$\exp(-M/2).dM.\exp(M/2) + dM = 0.$$

Let us rewrite this equation in an orthonormal basis in which $M$ is diagonal with a rotation matrix $R$. Let $(\lambda_i)$ be the eigenvalues of $M$ and let $dN := R.dM.R^T$. Then we have

$$dN = -\mathrm{Diag}(\exp(-\lambda_1/2), \ldots, \exp(-\lambda_N/2)).dN.\mathrm{Diag}(\exp(\lambda_1/2), \ldots, \exp(\lambda_N/2)).$$

Coordinate by coordinate, this is written as:

$$\forall i, j : dN_{i,j}(1 + \exp(-\lambda_i/2 + \lambda_j/2)) = 0.$$

Hence for all $i, j$ : $dN_{i,j} = 0$ which is equivalent to $dM = 0$. We are done. □

COROLLARY 2.9. [illegible] $\alpha \in \mathbb{R}$ [illegible] $S \mapsto S^\alpha$ [illegible] $\alpha = -1$

[illegible]. We have $S^\alpha = \exp(\alpha \log(S))$. The composition of smooth mappings is smooth. □

## 3. Log-Euclidean means.

In this section we focus on the construction of Log-Euclidean means. They are derived from two new structures on SPD matrices.

The first is a Lie group structure [11], i.e., an algebraic group structure that is [illegible] with the differential structure of the Space of SPD matrices. The second structure is a [illegible]. Indeed, one can define a [illegible] that complements the Lie group structure to form a vector space structure on the space of SPD matrices. In this context, Log-Euclidean metrics are defined as bi-invariant metrics on the Lie group of SPD matrices. The Log-Euclidean mean is the Fréchet mean associated with these metrics. It is particularly simple to compute.

### 3.1. Multiplication of SPD matrices.

It is not a priori obvious how one could define a multiplication on the space of SPD matrices compatible with classical algebraic and differential properties. How can one combine smoothly two SPD matrices to make a third one, in such a way that Id is still the identity and the usual inverse remains its inverse? Moreover, if we obtain a new Lie group structure, we would also like the matrix exponential to be the exponential associated with the Lie group structure which, a priori, can be different.

The first idea that comes to mind is to directly use matrix multiplication. But then the noncommutativity of matrix multiplication between SPD matrices stops the attempt: if $S_1, S_2 \in Sym_\star^+(n)$, $S_1.S_2$ is an SPD matrix (or equivalently, is symmetric) if and only if $S_1$ and $S_2$ commute. To overcome the possible asymmetry of the matrix product of two SPD matrices, one can simply take the [illegible] (i.e., the closest symmetric matrix in the sense of the Frobenius norm [7]) of the product and define the new product $\diamond$:

$$S_1 \diamond S_2 := \frac{1}{2}(S_1.S_2 + S_2.S_1).$$

This multiplication is smooth and conserves the identity and the inverse. But $S_1 \diamond S_2$ is not necessarily positive! Also, since the set of SPD matrices is not closed, one cannot define in general a closest SPD matrix, but only a closest symmetric [illegible] matrix [7].

In [12], affine-invariant distances between two SPD matrices $S_1, S_2$ are of the form

$$d(S_1, S_2) = \| \log(S_1^{-1/2}.S_2.S_1^{-1/2}) \|, \tag{3.1}$$

where $\|.\|$ is a Euclidean norm defined on $Sym(n)$. Let us define the following multiplication $\circledcirc$:

$$S_1 \circledcirc S_2 := S_1^{1/2}.S_2.S_1^{1/2}.$$

With this multiplication, the affine-invariant metric constructed in [12] can be interpreted then as a left-invariant metric. Moreover, this multiplication is smooth and compatible with matrix inversion and matrix exponential, and the product truly defines an SPD matrix. Everything works fine, except that it is not [illegible]. This makes everything fail, because associativity is an essential requirement of group structure. Without it, many fundamental properties disappear. For Lie groups, the notion of adjoint representation no longer exists without associativity.

Theorem 2.8 points to an important fact: $Sym_\star^+(n)$ is diffeomorphic to its tangent space at the identity, $Sym(n)$. But $Sym(n)$ has an additive group structure, and to obtain a group structure on the space of SPD matrices, one can simply transport the additive structure of $Sym(n)$ to $Sym_\star^+(n)$ with the exponential. More precisely, we have the following.

DEFINITION 3.1. [illegible] $S_1, S_2 \in Sym_\star^+(n)$ [illegible] $S_1 \odot S_2$ [illegible]

$$S_1 \odot S_2 := \exp(\log(S_1) + \log(S_2)). \tag{3.2}$$

PROPOSITION 3.2. $(Sym_\star^+(n), \odot)$ [illegible]

[illegible] The multiplication is defined by addition on logarithms. It is therefore associative and commutative. Since $\log(\mathrm{Id}) = 0$, the neutral element is Id, and since $\log(S^{-1}) = -\log(S)$, the new inverse is the matrix inverse. Finally, we have $\exp(\log(S_1) + \log(S_2)) = \exp(\log(S_1)).\exp(\log(S_2)) = S_1.S_2$ when $[S_1, S_2] = 0$. □

THEOREM 3.3. [illegible] $\odot$ [illegible] $Sym_\star^+(n)$ [illegible] $(S_1, S_2) \mapsto S_1 \odot S_2^{-1}$ [illegible] $\mathcal{C}^\infty$ [illegible] $Sym_\star^+(n)$ [illegible] $\odot$

[illegible] $(S_1, S_2) \mapsto S_1 \odot S_2^{-1} = \exp(\log(S_1) - \log(S_2))$. But since exp and log and the addition are smooth, their composition is also smooth. By definition (see [8, page 29]), $Sym_\star^+(n)$ is a Lie group. □

PROPOSITION 3.4. $\exp : (Sym(n), +) \rightarrow (Sym_\star^+(n), \odot)$ [illegible] $Sym_\star^+(n)$ [illegible] $Sym(n)$ [illegible] $(t.V)_{t\in\mathbb{R}}$ [illegible] $V \in Sym(n)$ [illegible] $Sym_\star^+(n)$ [illegible] $Sym(n)$

[illegible] We have explicitly transported the group structure of $Sym(n)$ into $Sym_\star^+(n)$ so exp is a morphism. It is also a bijection, and thus an isomorphism. The smoothness of exp then ensures its compatibility with the differential structure.

Let us recall the definition of one-parameter subgroups. $(S(t))_{t\in\mathbb{R}}$ is such a subgroup if and only if we have for all $t, s$: $S(t + s) = S(t) \odot S(s) = S(s) \odot S(t)$. But then $\log(S(t+s) = \log(S(t) \odot S(s)) = \log(S(t)) + \log(S(s))$ by definition of $\odot$. Therefore $\log S(t)$ is also a one-parameter subgroup of $(Sym(n), +)$, which is necessarily of the form $t.V$, where $V \in Sym(n)$. $V$ is the [illegible] of $S(t)$. Finally, the exponential is obtained from one-parameter subgroups, which are all of the form $(\exp(t.V))_{t\in\mathbb{R}}$ (see [5, Chap. V]). □

Thus, we have given the space of SPD matrices a structure of Lie group that leaves unchanged the classical matrix notions of inverse and exponential. The new multiplication used, i.e., the logarithmic multiplication, generalizes the matrix multiplication when two SPD matrices do not commute in the matrix sense.

The associated Lie algebra is the space of symmetric matrices, which is diffeomorphic and isomorphic to the group itself. The associated Lie bracket is the null bracket: $[S_1, S_2] = 0$ for all $S_1, S_2 \in Sym(n)$.

The reader should note that this Lie group structure is, to our knowledge, new in the literature. For a space as commonly used as SPD matrices, this is quite surprising. The probable reason is that the Lie group of SPD matrices is [illegible] a multiplicative matrix group, contrary to most Lie groups.

**3.2. Log-Euclidean metrics on the Lie group of SPD matrices.** Now that we have given $Sym_\star^+(n)$ a Lie group structure, we turn to the task of exploring metrics compatible with this new structure. Among Riemannian metrics in Lie groups, [illegible] metrics are the most convenient. We have the following definition.

DEFINITION 3.5. [illegible] $\langle , \rangle$ [illegible] $G$ [illegible] $m \in G$ [illegible] $m$ [illegible]

THEOREM 3.6. [illegible] [5, [illegible] V], [illegible]

1. [illegible]
2. [illegible] $m \in G$, $Ad(m)$ [illegible] $\mathfrak{g}$ [illegible] $Ad(m)$ [illegible] $m$
3. [illegible] $G$ [illegible]

COROLLARY 3.7. [illegible] $\langle , \rangle$ [illegible] $T_{Id}Sym_\star^+(n) = Sym(n)$ [illegible] $Sym_\star^+(n)$ [illegible]

[illegible]. The commutativity of the multiplication implies that $Ad(Sym_\star^+(n)) = \{\mathrm{Id}\}$, which is trivially an isometry group. ◻

This result is striking. In general Lie groups, the existence of bi-invariant metrics is not guaranteed. More precisely, it is guaranteed if and only if the adjoint representation $Ad(G)$ is relatively compact, i.e., (the dimension is assumed finite) if the group of matrices given by $Ad(G)$ is [illegible] (see [5, Theorem V.5.3]). This is trivially the case when the group is commutative, as here, since $Ad(G) = \{e\}$, which is obviously bounded. Other remarkable cases where $Ad(G)$ is bounded are compact groups, such as rotations. But for noncompact noncommutative groups, there is in general [illegible] bi-invariant metric, as in the case of rigid transformations.

DEFINITION 3.8. [illegible] 3.9

COROLLARY 3.9. [illegible] $\langle , \rangle$ [illegible] $Sym_\star^+(n)$ [illegible]

$$(\exp(V_1 + t.V_2))_{t\in\mathbb{R}}, \quad [illegible] \quad V_1,\ V_2 \in Sym(n). \tag{3.3}$$

[illegible]

$$\begin{cases} \log_{S_1}(S_2) = D_{\log(S_1)} \exp .(\log(S_2) - \log(S_1)), \\ \exp_{S_1}(L) = \exp(\log(S_1) + D_{S_1} \log .L). \end{cases} \tag{3.4}$$

[illegible] $V_1$, $V_2$ [illegible] $S$ [illegible]

$$\langle V_1, V_2 \rangle_S = \langle D_S \log .V_1, D_S \log .V_2 \rangle_{Id}. \tag{3.5}$$

[illegible]

$$d(S_1, S_2) = \| \log_{S_1}(S_2) \|_{S_1} = \| \log(S_2) - \log(S_1) \|_{Id}, \tag{3.6}$$

[illegible] $\|.\|$ [illegible]

[illegible]. Theorem 3.6 states that geodesics are obtained by translating one-parameter subgroups, and Proposition 3.4 gives the form of these subgroups in terms of the matrix exponential. By definition, the metric exponential $\exp_{S_1} : T_{S_1}Sym_\star^+(n) \to Sym_\star^+(n)$ is the mapping that associates with a tangent vector $L$ the value at time 1 of the geodesic starting at time 0 from $S_1$ with an initial speed vector $L$. Differentiating the geodesic equation (3.3) at time 0 yields an initial vector speed equal to

$D_{V_1}\exp.V_2$. As a consequence, $\exp_{S_1}(L) = \exp(\log(S_1) + (D_{\log(S_1)}\exp)^{-1}.L)$. The differentiation of the equality $\log\circ\exp = \mathrm{Id}$ yields $(D_{\log(S_1)}\exp)^{-1} = D_{S_1}\log$. Hence we have the formula for $\exp_{S_1}(L)$. Solving in $L$ the equation $\exp_{S_1}(L) = S_2$ provides the formula for $\log_{S_1}(S_2)$.

The metric at a point $S$ is obtained by propagating by translation the scalar product on the tangent space at the identity. Let $L_S : Sym_\star^+(n) \to Sym_\star^+(n)$ be the logarithmic multiplication by $S$. We have $\langle V_1, V_2\rangle_S = \langle D_S L_{S^{-1}}.V_1, D_S L_{S^{-1}}.V_2\rangle$. But simple computations show that $D_S L_{S^{-1}} = D_S\log$. Hence we have (3.5). Finally, we combine (3.4) and (3.5) to obtain the (simple this time!) formula for the distance. □

COROLLARY 3.10. [illegible] ( [8, [illegible] 107]) [illegible] $Sym(n)$ [illegible]

In [12], the metric defined on the space of SPD matrices is affine invariant. The action act$(A)$ of an invertible matrix $A$ on the space of SPD matrices is defined by

$$\forall S,\ \mathrm{act}(A)(S) = A.S.A^T.$$

Affine-invariance means that for all invertible matrices $A$, the mapping $\mathrm{act}(A) : Sym_\star^+(n) \to Sym_\star^+(n)$ is an isometry. This group action describes how an SPD matrix, assimilated to a covariance matrix, is affected by a general affine change of coordinates.

Here, the Log-Euclidean Riemannian framework will not yield full affine-invariance. However, it is not far from it, because we can obtain invariance by similarity (isometry plus scaling).

PROPOSITION 3.11. [illegible] $Sym_\star^+(n)$ [illegible] $\langle V_1, V_2\rangle :=$ [illegible] $(V_1.V_2)$ [illegible] $V_1, V_2 \in Sym(n)$

[illegible] Let $R \in SO(n)$ be a rotation and $s > 0$ be a scaling factor. Let $S$ be an SPD matrix. $V$ is transformed by the action of $s.R$ into $\mathrm{act}(sR)(S) = s^2.R.S.R^T$. From (3.6), the distance between two SPD matrices $S_1$ and $S_2$ transformed by $sR$ is

$$d(\mathrm{act}(sR)(S_1), \mathrm{act}(sR)(S_2)) = \mathrm{Trace}(\{\log(\mathrm{act}(sR)(S_1)) - \log(\mathrm{act}(sR)(S_2))\}^2).$$

A scaling by a positive factor $\lambda$ on an SPD matrix corresponds to a translation by $\log(\lambda).\mathrm{Id}$ in the domain of logarithms. Furthermore, we have $\log(R.S.R^T) = R.\log(S).R^T$ for any SPD matrix $S$ and any rotation $R$. Consequently, the scaling zeros out in the previous formula and we have

$$\begin{aligned} d(\mathrm{act}(sR)(S_1), \mathrm{act}(sR)(S_2)) &= \mathrm{Trace}(\{R.(\log(S_1) - \log(S_2)).R^T\}^2) \\ &= \mathrm{Trace}(\{\log(S_1) - \log(S_2)\}^2) \\ &= d(S_1, S_2). \end{aligned}$$

Hence we have the result. □

Thus, we see that the Lie group of SPD matrices with an appropriate Log-Euclidean metric has many [illegible]: Lie group bi-invariance and similarity-invariance. Moreover, Theorem 3.6 shows that the inversion mapping $S \mapsto S^{-1}$ is an isometry.

**3.3. A vector space structure on SPD matrices.** We have already seen that the Lie group of SPD matrices is isomorphic and diffeomorphic to the additive group of symmetric matrices. We have also seen that with a Log-Euclidean metric, the Lie group of SPD matrices is also isometric to the space of symmetric matrices endowed with the associated Euclidean metric. There is more: the Lie group isomorphism exp from the Lie algebra of symmetric matrices to the space of SPD matrices can be smoothly extended into an isomorphism of vector spaces. Indeed, let us define the following operation.

DEFINITION 3.12. [illegible] $\circledast$ [illegible] $\lambda \in \mathbb{R}$ [illegible]

$$\lambda \circledast S = \exp(\lambda . \log(S)) = S^{\lambda}. \tag{3.7}$$

When we assimilate the logarithmic multiplication to an addition and the logarithmic scalar multiplication to a usual scalar multiplication, we have all the properties of a vector space. By construction, the mapping $\exp : (Sym(N), +, .) \to (Sym_{\star}^{+}(n), \odot, \circledast)$ is a vector space isomorphism. Since all algebraic operations on this vector space are smooth, this defines what could be called a "Lie vector space structure" on SPD matrices.

Of course, this result does not imply that the space of SPD matrices is a vector subspace of the vector space of square matrices. But it shows that we can view this space as a vector space when we identify an SPD matrix with its logarithm. The question of whether or not the SPD matrix space is a vector space depends on the vector space structure we are considering, and [illegible] on the space itself.

From this point of view, bi-invariant metrics on the Lie group of SPD matrices are simply the classical Euclidean metrics on the vector space $(Sym(n), +, .)$. Thus, we have in fact defined a new Euclidean structure on the space of SPD matrices by transporting that of its Lie algebra $Sym(n)$ on SPD matrices. But this Euclidean structure does not have the defects mentioned in the introduction of this article: matrices with null eigenvalues are at infinite distance and the symmetry principle is respected. Last but not least, with an appropriate metric, similarity-invariance is also guaranteed.

**3.4. Log-Euclidean mean.** We present here the definition of the Log-Euclidean mean of SPD matrices and its invariance properties.

THEOREM 3.13. [illegible] $(S_i)_1^N$ [illegible]

$$\mathbb{E}_{LE}(S_1, \ldots, S_N) = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log(S_i)\right). \tag{3.8}$$

[illegible]

[illegible]. When one expresses distances in the logarithm domain, one is faced with the classical computation of an Euclidean mean. Hence we have the formula by mapping back the results with exp in the domain of SPD matrices. Now, this mean does not depend on the chosen Log-Euclidean metric, and since there exist similarity-invariant metrics among Log-Euclidean metrics, this property propagates to the mean. The three last invariance properties are reformulations in the domain of SPD matrices of classical properties of the arithmetic mean in the domain of logarithms. ☐

Table 4.1

*Comparison between affine-invariant and Log-Euclidean metrics. Note on the one hand the important simplifications in terms of distance and geodesics in the Log-Euclidean case. On the other hand, this results in the use of the differentials of the matrix exponential and logarithm in the exponential and logarithm maps.*

| Affine-invariant metrics | Log-Euclidean metrics |
|---|---|
| Exponential map: $\exp_{S_1}(L) =$ | |
| $S_1^{1/2}.\exp(S_1^{-1/2}.L.S_1^{-1/2}).S_1^{1/2}$ | $\exp(\log(S_1) + D_{S_1}\log .L)$ |
| Logarithm map: $\log_{S_1}(S_2) =$ | |
| $S_1^{1/2}.\log(S_1^{-1/2}.S_2.S_1^{-1/2}).S_1^{1/2}$ | $D_{\log(S_1)}\exp .(\log(S_2) - \log(S_1))$ |
| Dot product: $\langle L_1, L_2\rangle_S =$ | |
| $\langle S^{-1/2}.L_1.S^{-1/2}, S^{-1/2}.L_2.S^{-1/2}\rangle_{\mathrm{Id}}$ | $\langle D_S\log .L_1, D_S\log .L_2\rangle_{\mathrm{Id}}$ |
| Distance: $d(S_1, S_2) =$ | |
| $\lVert \log(S_1^{-1/2}.S_2.S_1^{-1/2})\rVert$ | $\lVert \log(S_2) - \log(S_1)\rVert$ |
| Geodesic between $S_1$ and $S_2$: | |
| $S_1^{1/2}.\exp(tW).S_1^{1/2}$<br>with $W = \log\left(S_1^{-1/2}.L.S_1^{-1/2}\right)$ | $\exp\left((1-t)\log(S_1) + t\log(S_2)\right)$ |
| Invariance properties | |
| Affine-invariance | Lie group bi-invariance,<br>Similarity-invariance |

**4. Comparison with the affine-invariant mean.** In this section we compare the Log-Euclidean mean to the recently introduced affine-invariant mean [12, 19, 23, 22]. To this end, we first recall the differences between affine-invariant metrics and Log-Euclidean metrics in terms of elementary operators, distance, and geodesics. Then we turn to a study of the algebraic properties of Fréchet means in the Log-Euclidean and affine-invariant cases.

**4.1. Elementary metric operations and invariance.** Distances, geodesics, and Riemannian means take a much simpler form in the Log-Euclidean than in the affine-invariant case. Invariance properties are comparable: some Log-Euclidean metrics are not only bi-invariant but also similarity invariant. These properties are summarized in Table 4.1. However, we see in this table that the exponential and logarithmic mappings are complicated in the Log-Euclidean case by the use of the differentials of the matrix exponential and logarithm. This is the price to pay to obtain simple distances and geodesics. Interestingly, using spectral properties of symmetric matrices, one can obtain a closed form for the differential of both matrix logarithm and exponential and it is possible compute them very efficiently. See [26] for more details.

**4.2. Affine-invariant means.** Let $(S_i)_{i=1}^N$ be a system of SPD matrices. Contrary to the Log-Euclidean case, there is in general no closed form for the affine-invariant Fréchet mean $E_{Aff}(S_1, \dots, S_N)$ associated with affine-invariant metrics. The affine-invariant mean is defined [illegible] by a [illegible], which is the following:

$$\sum_{i=1}^{N} \log(\mathbb{E}_{Aff}(S_1, \dots, S_N)^{-1/2}.S_i.\mathbb{E}_{Aff}(S_1, \dots, S_N)^{-1/2}) = 0. \tag{4.1}$$

This equation is equivalent to the following other barycentric equation, given in [19]:

$$\sum_{i=1}^{N} \log(\mathbb{E}_{Aff}(S_1, \ldots, S_N)^{-1}.S_i) = 0. \tag{4.2}$$

The two equations are equivalent simply because for all $i$,

$$\mathbb{E}_{Aff}(S_1, \ldots, S_N)^{-1/2}.S_i.\mathbb{E}_{Aff}(S_1, \ldots, S_N)^{-1/2} = A.\mathbb{E}_{Aff}(S_1, \ldots, S_N)^{-1}.S_i.A^{-1}$$

with $A = \mathbb{E}_{Aff}(S_1, \ldots, S_N)^{-1/2}$. The fact that $\log(A.S.A^{-1}) = A.\log(S).A^{-1}$ suffices to conclude.

To solve (4.1), the only known strategy is to resort to an iterative numerical procedure, such as the Gauss–Newton gradient descent method described in [12].

**4.3. Geometric interpolation of determinants.** The definition of the Log-Euclidean mean given by (3.8) is extremely similar to that of the classical scalar geometrical mean. We have the following classical definition.

DEFINITION 4.1. [illegible] $d_1, \ldots, d_N$ [illegible]

$$\mathbb{E}(d_1, \ldots, d_N) = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log(d_i)\right).$$

The Log-Euclidean and affine-invariant Fréchet means can [illegible] be considered as generalizations of the geometric mean. Indeed, their determinants are both equal to the scalar geometric mean of the determinants of the original SPD matrices. This fundamental property can be thought of as the [illegible] to SPD matrices.

THEOREM 4.2. [illegible] $(S_i)_{i=1}^{N}$ [illegible] $N$ [illegible]

[illegible] From Proposition 2.4 we know that $\det(\exp(M)) = \exp(\text{Trace}(M))$ for any square matrix $M$. Then for the geometric mean, we get

$$\begin{aligned}
\det(\mathbb{E}_{LE}(S_1, \ldots, S_N)) &= \exp(\text{Trace}(\log(\mathbb{E}_{LE}(S_1, \ldots, S_N)))) \\
&= \exp\left(\text{Trace}\left(\frac{1}{N}\sum_{i=1}^{N} \log(S_i)\right)\right) \\
&= \exp\left(\frac{1}{N}\sum_{i=1}^{N} \log(\det(S_i))\right) \\
&= \exp\left(\mathbb{E}(\log(\det(S_1, \ldots, S_N)))\right).
\end{aligned}$$

For affine-invariant means, there is no closed form for the mean. [illegible] there is the barycentric equation given by (4.1). By applying the same formula as before after having taken the exponential and using $\det(S.T) = \det(S).\det(T)$ we obtain the result. □

Theorem 4.2 shows that the Log-Euclidean and affine-invariant means of SPD matrices are quite similar. In terms of interpolation, this result is satisfactory, since it implies that the interpolated determinant, i.e., the volume of the associated interpolated ellipsoids, will vary between the values of the determinants of the source SPD matrices. Indeed, we have the following.

Corollary 4.3. [illegible] $(S_i)_{i=1}^N$ [illegible] $N$ [illegible]

$$\left[\inf_{i\in 1\dots N}(S_i),\ \sup_{i\in 1\dots N}(S_i)\right].$$

[illegible]. This is simply a consequence of the monotonicity of the scalar exponential and of the scalar integral. □

Corollary 4.4. [illegible] $S_1$ [illegible] $S_2$ [illegible]

[illegible]. Indeed, in both cases, the interpolated determinant $\mathrm{Det}(t)$ is the geometric mean of the two determinants, i.e., at $t \in [0,1]$: $\mathrm{Det}(t) = \exp((1-t)\log(\det(S_1)) + t\log(\det(S_2)))$. This interpolation is monotonic, since the differentiation yields

$$\frac{d}{dt}\mathrm{Det}(t) = \mathrm{Det}(t)\log(\det(S_2.S_1^{-1})).$$

As a consequence, $\mathrm{Det}(t)$ is equal to $\det(S_1).\exp(t.\log(\det(S_2.S_1^{-1})))$, and the sign of $\frac{d}{dt}\mathrm{Det}(t)$ is constant and given by $\log(\det(S_2.S_1^{-1}))$. □

**4.4. Criterion for the equality of the two means.** In general, Log-Euclidean and affine-invariant means are similar, yet they are [illegible] identical. Nonetheless, there are a number of cases where they are identical, for example, when the logarithms of averaged SPD matrices all commute with one another. In fact, we have more as follows.

Proposition 4.5. [illegible] $(S_i)_{i=1}^N$ [illegible] $N$ [illegible] $\log(S_i)$ [illegible]

[illegible]. Let $\bar{L} := \frac{1}{N}\sum_{i=1}^N \log(S_i)$. The hypothesis is that $[\bar{L}, \log(S_i)] = 0$ for all $i$. This implies that $\log(\exp(-\frac{1}{2}\bar{L}).S_i.\exp(-\frac{1}{2}\bar{L})) = \log(S_i) - \bar{L}$ for all $i$. We see then that $\exp\bar{L}$, i.e., the Log-Euclidean mean, is the solution of (4.1), i.e., is the affine-invariant mean. □

So far, we have not been able to prove the converse part of this proposition. However, the next subsection provides a partial proof, valid when SPD matrices are isotropic enough, i.e., close to a scaled version of the identity. The intensive numerical experiments we have carried out strongly suggest that the result given in the next section is true in general. The full proof of this assertion will be the subject of future work.

**4.5. Larger anisotropy in Log-Euclidean means.** In section 4.6, we will verify experimentally that affine-invariant means tend to be less anisotropic than Log-Euclidean means. The following theorem accounts for this phenomenon when SPD matrices are isotropic enough.

Theorem 4.6. [illegible] $(S_i)_{i=1}^N$ [illegible] 2 [illegible] $\log(S_i)$ [illegible]

$$\text{[illegible]}\,(\mathbb{E}_{Aff}(S_1,\dots,S_N)) < \text{[illegible]}\,(\mathbb{E}_{LE}(S_1,\dots,S_N)). \tag{4.3}$$

[illegible]. The idea is to see how the two means differ close to the identity. To this end, we introduce a small scaling factor $t$ and see how the two means vary when $t$ is close to zero. For all $i$, let $S_{i,t}$ be the version of $S_i$ scaled by $t$ in the logarithmic domain. Around the identity, we can use the Baker–Campbell–Hausdorff formula to simplify the barycentric equation (4.1). Let us denote both Riemannian cases as $\mathbb{E}(S_t) = \mathbb{E}(S_{1,t}, \ldots, S_{N,t})$ and $\mathbb{E}(S) := \mathbb{E}(S_1, \ldots, S_N)$. We will also use the following notation: $\log(S_i) := L_i$, $\bar{L}_{t;Aff} := \log(\mathbb{E}_{Aff}(S_t))$ and $\bar{L}_{LE} := \log(\mathbb{E}_{LE}(S))$.

%pagebreak

First, we use twice the Baker–Campbell–Hausdorff formula to obtain the following approximation:

$$\begin{aligned}\log(\mathbb{E}_{Aff}(S_t)^{-1/2}.S_{i,t}.\mathbb{E}_{Aff}(S_t)^{-1/2}) = \;& tL_i - \bar{L}_{t;Aff} - t^3\tfrac{1}{12}[L_i,[L_i,\bar{L}_{t;Aff}]] \\ & + t^3\tfrac{1}{24}[\bar{L}_{t;Aff},[\bar{L}_{t;Aff},L_i]] + O(t^5).\end{aligned} \tag{4.4}$$

Then we average over $i$ to obtain the following approximation lemma.

Lemma 4.7. [illegible] $t$ [illegible]

$$\bar{L}_{t;Aff} = t\bar{L}_{LE} + \frac{t^3}{12.N}\sum_{i=1}^{N}[L_i,[\bar{L}_{LE},L_i]] + O(t^5). \tag{4.5}$$

[illegible]. To obtain the approximation, note that the second factor $t^3\frac{1}{24}[\bar{L}_{t;Aff}, [\bar{L}_{t;Aff}, L_i]]$ in (4.4) becomes an $O(t^5)$. Indeed, when the sum over $i$ is done, $L_i$ becomes $\bar{L}_{LE}$. But we can replace $\bar{L}_{LE}$ with its value in term of the affine-invariance mean by using (4.4). Then, using the fact that $[\bar{L}_{t;Aff}, \bar{L}_{t;Aff}] = 0$ we see that we obtain an $O(t^5)$.

Note also that, thanks to the symmetry with respect to inversion, $\bar{L}_{t;Aff}$ becomes $-\bar{L}_{t;Aff}$ when $t$ is changed into $-t$, i.e., $t \mapsto \bar{L}_{t;Aff}$ is odd. As a consequence, only odd terms appear in the development in powers of $t$. ☐

Next, we take the exponential of (4.5) and differentiate the exponential to obtain

$$\mathbb{E}_{Aff}(S_t) = \mathbb{E}_{LE}(S_t) + D_{t\bar{L}_{LE}}\exp.\left(\frac{t^3}{12.N}\sum_{i=1}^{N}[L_i,[\bar{L}_{LE},L_i]]\right) + O(t^5).$$

Then we use several properties to approximate the trace of affine-invariant means. First, we use Corollary 2.3 to simplify the use of the differential of the exponential. Then we approximate the exponential by the first two terms of its series expansion. We obtain

$$\mathrm{Trace}(\mathbb{E}_{Aff}(S_t)) = \mathrm{Trace}(\mathbb{E}_{LE}(S_t)) + t^3.F(t,L_i,\bar{L}_{LE}) + O(t^5),$$

with $F(t,L_i,\bar{L}_{LE}) = \mathrm{Trace}(\exp(t\bar{L}_{LE}).\frac{1}{12.N}\sum_{i=1}^{N}[L_i,[\bar{L}_{LE},L_i]])$. This expression can be simplified as follows:

$$\begin{aligned}F(t,L_i,\bar{L}_{LE}) &= \mathrm{Trace}\left((\mathrm{Id}+t\bar{L}_{LE}).\frac{1}{12.N}\sum_{i=1}^{N}[L_i,[\bar{L}_{LE},L_i]]\right) + O(t^2) \\ &= \frac{t}{12.N}\sum_{i=1}^{N}\mathrm{Trace}\left(\bar{L}_{LE}.[L_i,[\bar{L}_{LE},L_i]]\right) + O(t^2) \\ &= -\frac{t}{12.N}\sum_{i=1}^{N}\mathrm{Trace}\left(L_i^2.\bar{L}_{LE}^2 - (L_i.\bar{L}_{LE})^2\right) + O(t^2).\end{aligned}$$

As a consequence, the difference between the two traces can be written as

$$\mathrm{Trace}(\mathbb{E}_{Aff}(S_t)) - \mathrm{Trace}(\mathbb{E}_{LE}(S_t)) = -\frac{t^4}{12.N}\sum_{i=1}^{N} \mathrm{Trace}\left(L_i^2.\bar{L}_{LE}^2 - (L_i.\bar{L}_{LE})^2\right) + O(t^5).$$

To conclude, we use the following lemma.

LEMMA 4.8. [illegible] $A$, $B \in Sym(n)$ [illegible] $(A^2.B^2 - (A.B)^2) \geq 0$ [illegible] $A$ [illegible] $B$ [illegible]

[illegible]. Let $(A_i)$ (resp., $(B_i)$) be the column vectors of $A$ (resp., $B$). Let $\langle , \rangle$ be the usual scalar product. Then we have

$$\begin{cases} \mathrm{Trace}(A^2.B^2) = \sum_{i,j}\langle A_i, A_j\rangle\langle B_i, B_j\rangle, \\ \mathrm{Trace}((A.B)^2) = \sum_{i,j}\langle A_i, B_j\rangle\langle B_i, A_j\rangle. \end{cases}$$

Let us now chose a rotation matrix $R$ that makes $A$ diagonal: $\mathrm{R.A.R^T} = Diag(\lambda_1, \dots, \lambda_n) =: D$. Let us define $C := R.B.R^T$ and use the notation $(C_i)$ and $(D_i)$ for the column vectors of $C$ and $D$. We have

$$\begin{cases} \mathrm{Trace}(A^2.B^2) = \sum_{i,j}\langle D_i, D_j\rangle\langle C_i, C_j\rangle = \sum_i \lambda_i^2\langle C_i, C_i\rangle, \\ \mathrm{Trace}((A.B)^2) = \sum_{i,j}\langle D_i, C_j\rangle\langle C_i, D_j\rangle = \sum_{i,j}\lambda_i.\lambda_j\langle C_i, C_j\rangle. \end{cases}$$

Then the Cauchy–Schwarz inequality yields

$$\left|\sum_{i,j}\lambda_i.\lambda_j\langle C_i, C_j\rangle\right| \leq \sum_i \lambda_i^2\langle C_i, C_i\rangle,$$

which proves the first point. But the Cauchy–Schwarz inequality is an equality if and only if there is a constant $\mu$ such that $D.C = \mu C.D$. But only $\mu = 1$ allows the inequality of the lemma to be an equality. This is equivalent to $C.D = D.C$, which is equivalent in turn to $A.B = B.A$. Hence we have the result. □

[illegible] 4.6 When we apply Lemma 4.8 to the obtained estimation for the trace, we see that for a $t \neq 0$ small enough, the trace of the affine-invariant mean is indeed strictly inferior to the trace of the Log-Euclidean mean whenever the mean logarithm does not commute with all logarithms $\log(S_i)$. □

COROLLARY 4.9. [illegible] 4.6 [illegible] $\lambda$ [illegible] $\lambda > 0$

COROLLARY 4.10. [illegible] 2 [illegible]

[illegible]. In this case, there are only two eigenvalues for each mean. Their products are equal and we have a strict inequality between their sums. Consequently, the largest eigenvalue of the Log-Euclidean mean is strictly larger than the affine-invariant one, and we have the opposite result for the smallest eigenvalue. □

**4.6. Linear and bilinear interpolation of SPD matrices.** Volume elements (or [illegible]) in clinical DT images are often spatially anisotropic. Yet, in many practical situations where DT images are used, it is recommended (see [27]) to work with isotropic voxels to avoid spatial biases. A preliminary resampling step with an adequate interpolation method is therefore important in many cases. Proper interpolation methods are also required to generalize to the SPD case usual registration

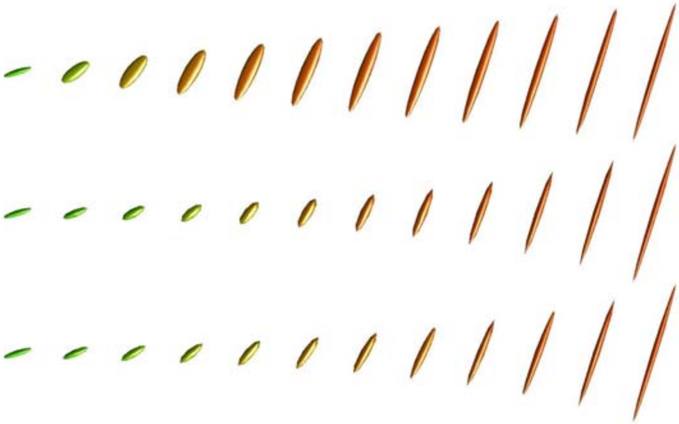


FIG. 4.1. *Linear interpolation of two SPD matrices. Top: linear interpolation on coefficients. Middle: affine-invariant interpolation. Bottom: Log-Euclidean interpolation. The shading of ellipsoids is based on the direction of dominant eigenvectors. Note the characteristic swelling effect observed in the Euclidean case, which is not present in both Riemannian frameworks. Note also that Log-Euclidean means are slightly more anisotropic their affine-invariant counterparts.*

techniques used on scalar or vector images. The framework of Riemannian metrics allows a direct generalization to SPD matrices of classical resampling methods with the use of associated Fréchet means instead of the Euclidean (i.e., arithmetic) mean.

In the Riemannian case, the equivalent of linear interpolation is [illegible]. To interpolate between two SPD matrices, intermediate values are taken along the shortest path joining the two matrices. Figure 4.1 presents a typical result of linear interpolation between two SPD matrices. The Euclidean, affine-invariant, and Log-Euclidean results are given. The "swelling effect" is clearly visible in the Euclidean case: the volume of associated ellipsoids is parabolically interpolated and reaches a global maximum between the two extremities! This effect disappears in both Riemannian cases, where volumes are interpolated geometrically. As expected, Log-Euclidean means are a little more anisotropic than their affine-invariant counterparts.

To resample images, bilinear (resp., trilinear) interpolation generalizes in two dimensions (resp., in three dimensions) the linear interpolation and offers an efficient compromise between simplicity and accuracy in the scalar and vector cases. With this technique, the value at any given point is inferred from known values measured at the vertices of a regular grid whose elementary cells are rectangles in two dimensions (resp., right parallelepipeds in three dimensions), which is usually the case with MR images. More precisely, the interpolated value at a given point is given by the weighted mean of the values at the vertices of the current cell. The weights are the [illegible] of the current point with respect to the vertices of the current cell.

Figure 4.2 presents the results of the bilinear interpolation of four SPD matrices placed at the extremities of a rectangle. Again, a large swelling effect is present in Euclidean results and not in both Riemannian results, and Log-Euclidean means are slightly more anisotropic than their affine-invariant equivalents. One should note that the computation of the affine-invariant mean here is iterative, since the number of averaged matrices is larger than 2 (we use the Gauss–Newton method described in [12]), whereas the closed form given by 3.8 is used directly in the Log-Euclidean case. This has a large impact on computation times: $0.003s$ (Euclidean), $0.009s$

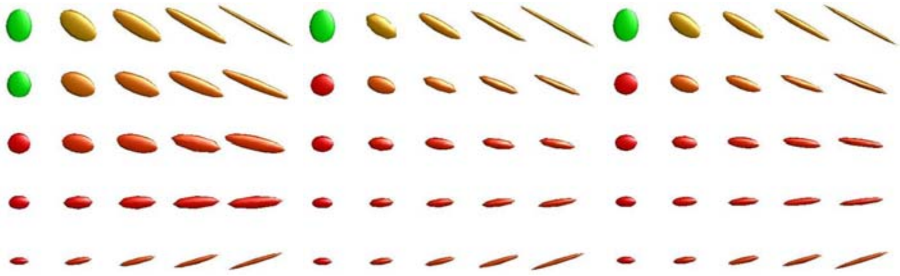


Fig. 4.2. *Bilinear interpolation of four SPD matrices at the corners of a regular grid. Left: Euclidean interpolation. Middle: affine-invariant interpolation. Right: Log-Euclidean interpolation. Again, a characteristic swelling effect is observed in the Euclidean case and not in both Riemannian frameworks. As expected, Log-Euclidean means are slightly more anisotropic than their affine-invariant counterparts.*

(Log-Euclidean), and $1s$ (affine-invariant) for a $5 \times 5$ grid on a Pentium M 2 GHz. Computations were carried out with MATLAB, which explains the poor computational performance. Here, Log-Euclidean means were calculated approximately 100 times faster than affine-invariant means because the logarithms of the four interpolated tensors were computed only once, instead of being computing each time a new barycenter is calculated. When only one mean is computed, the typical ratio is closer to 20, since between 15 and 20 iterations are typically needed (for $3 \times 3$ SPD matrices) to obtain the affine-invariant mean with a precision of the order of $10^{-12}$.

One should note that from a numerical point of view the computation of Log-Euclidean means is not only much faster but also more [illegible] than in the affine-invariant case. On synthetic examples, as soon as SPD matrices are quite anisotropic (for instance, with the dominant eigenvalue larger than 500 times the smallest), numerical instabilities appear, essentially due to limited numerical precision (even with double precision). This can greatly complicate the computation of affine-invariant means. On the contrary, the computation of Log-Euclidean means is more stable since the logarithm and exponential are taken only once and thus even very large anisotropies can be dealt with. In applications where very high anisotropies are present, such as the generation of adapted meshes [17], this phenomenon could severely limit the use of affine-invariant means, whereas no such limitation exists in the Log-Euclidean case.

**5. Conclusion and perspectives.** In this work, we have presented a particularly simple and efficient generalization of the geometric mean to SPD matrices, called Log-Euclidean. It is simply an arithmetic mean in the domain of matrix logarithms. This mean corresponds to a bi-invariant mean in our novel Lie group structure on SPD matrices, or equivalently to a Euclidean mean when this structure is smoothly extended into a vector space by a novel scalar multiplication.

The Log-Euclidean mean is similar to the recently introduced affine-invariant mean, which is another generalization of the geometric mean to SPD matrices. Indeed, the Log-Euclidean mean is similarity invariant, and two means have the same determinant, which is the geometric mean of the determinants of averaged SPD matrices. However, they are not equal: the Log-Euclidean trace is larger when the two means differ. The most striking difference between the two means resides in their computational cost: the Log-Euclidean mean can be calculated approximately 20 times faster than the affine-invariant mean. This property can be crucial in applications where large amounts of data are processed. This is especially the case in medical imaging with DTI and in numerical analysis with the generation of adapted meshes.

We have shown in this work that there are indeed several generalizations of the geometric mean to SPD matrices. Other variants may exist, and we will investigate other possible generalizations in future work. This is important, since situations in applied mathematics, mechanics, medical imaging, etc., where SPD matrices need to be processed, are highly varied. As a consequence, the relevance of each generalization of the geometric mean and of the associated metric framework may depend on the application considered. We have already begun to compare the Log-Euclidean and affine-invariant frameworks in the case of DT-MRI processing [28]. In future work, we will proceed to variability tensors, which we began to use in [15] to model and analyze the variability of brain anatomy.

## REFERENCES

[1] S. Lang, *Algebra*, 3rd rev. ed. Grad. Texts in Math., 211 Springer-Verlag, New York, 2002.
[2] M. L. Curtis, *Matrix Groups*, Springer-Verlag, New York, Heidelberg, 1979.
[3] D. Le Bihan, *Diffusion MNR imaging*, Magnetic Resonance Quarterly, 7 (1991), pp. 1–30.
[4] J. Salencon, *Handbook of Continuum Mechanics*, Springer-Verlag, Berlin, 2001.
[5] S. Sternberg, *Lectures on Differential Geometry*, Prentice–Hall, Englewood Cliffs, NJ, 1964.
[6] L. Schwartz, *Analyse Tome* 2: *Calcul Differentiel*, Hermann, Paris, 1997.
[7] N. J. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J. C. Gover and S. Barnett, eds., Oxford University Press, Oxford, UK, 1989, pp. 1–27.
[8] S. Gallot, D. Hulin, and J. Lafontaine, *Riemannian Geometry*, 2nd ed., Springer-Verlag, Berlin, 1990.
[9] R. Godement, *Introduction à la Théorie des Groupes de Lie*, Publications Mathématiques de l'Université Paris VII, Paris, 1982.
[10] M. Wüstner, *A connected lie group equals the square of the exponential image*, J. Lie Theory, 13 (2003), pp. 307–309.
[11] N. Bourbaki, *Elements of Mathematics: Lie Groups and Lie Algebra. Chapters* 1–3, Springer-Verlag, Berlin, 1989.
[12] X. Pennec, P. Fillard, and N. Ayache, *A Riemannian framework for tensor computing*, International J. Computer Vision, 66 (2006), pp. 41–66. A preliminary version appeared as Research Report 5255, INRIA, Sophia-Antipolis, France, 2004.
[13] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 1179–1193.
[14] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, Grad. Texts in Math., Springer-Verlag, New York, 2003.
[15] P. Fillard, V. Arsigny, X. Pennec, K. M. Hayashi, P. M. Thompson, and N. Ayache, *Measuring brain variability by extrapolating sparse tensor fields measured on sulcal lines*, Neuroimage, 34 (2007), pp. 639–650.
[16] T. Broxand, M. Roussonand, R. Deriche, and J. Weickert, *Unsupervised segmentation incorporating colour, texture, and motion*, in Computer Analysis of Images and Patterns N. Petkov and M.A. Westenbere, eds., Lecture Notes in Comput. Sci. 2756, Springer, Berlin, 2003, pp. 353–360.
[17] B. Mohammadi, H. Borouchaki, and P. L. George, *Delaunay mesh generation governed by metric specifications. II. Applications*, Finite Elem. Anal. Des. as, (1997), pp. 85–109.
[18] X. Pennec, *Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements*, J. Math. Imaging Vision, 25 (2006), pp. 127–154. A preliminary version appeared as Research Report RR-5093, INRIA, Sophia-Antipolis, France, 2004.
[19] M. Moakher, *A differential* geometry *approach to the geometric mean of symmetric positive-definite matrices*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 735–747.
[20] C. Feddern, J. Weickert, B. Burgeth, and M. Welk, *Curvature-driven PDE methods for matrix-valued images*, Internat. J. Comput. Vision, 69 (2006), pp. 91–103. Revised version of Tech. Report 104, Department of Mathematics, Saarland University, Saarbrücken, Germany, 2004.
[21] C. Chefd'hotel, D. Tschumperlé, R. Deriche, and O. Faugeras, *Regularizing flows for constrained matrix-valued images*, J. Math. Imaging Vision, 20 (2004), pp. 147–162.
[22] P.T. Fletcher and S.C. Joshi, *Principal geodesic analysis on symmetric spaces: Statistics*

*of diffusion tensors.*, in Proceedings of the CVAMIA and MMBIA Workshops, (Prague, Czech Republic, May 15, 2004), Lecture Notes in Comput. Sci. 3117, Springer, Berlin, 2004, pp. 87–98.

[23] C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras, *Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor MRI processing*, J. Math. Imaging Vision, 25 (2006), pp. 423–444.

%pagebreak

[24] S. Hun Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub, *Approximating the logarithm of a matrix to specified accuracy*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1112–1125.

[25] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, *Fast and Simple Computations on Tensors with Log-Euclidean Metrics*, Research Report RR-5584, INRIA, Sophia-Antipolis, France, 2005.

[26] P. Fillard, V. Arsigny, X. Pennec, and N. Ayache, *Clinical DT-MRI estimation, smoothing and fiber tracking with log-Euclidean metrics*, in Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006), Arlington, Virginia, 2006, pp. 786–789.

[27] P. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, *In vivo fiber tractography using DT-MRI data*, Magnetic Resonance in Medicine, 44 (2000), pp. 625–632.

[28] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, *Log-Euclidean metrics for fast and simple calculus on diffusion tensors*, Magnetic Resonance in Medicine, 56 (2006), pp. 411–421.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 349–369

# CONVERGENCE OF A BLOCK-ORIENTED QUASI-CYCLIC JACOBI METHOD*

VJERAN HARI†

**Abstract.** This paper proves the global convergence of a block-oriented, quasi-cyclic Jacobi method for symmetric matrices. The result applies to the new fast one-sided Jacobi method, proposed by Drmač and Veselić, for computing the singular value decomposition. There is no restriction on the matrix block-partition which defines the pivot strategy.

**Key words.** eigenvalues, Jacobi method, global convergence

**AMS subject classification.** 65F15

**DOI.** 10.1137/05064552X

## 1. Introduction.

Recently, Drmač and Veselić [5, 6] have proposed an improved modification of the one-sided Jacobi method for computing the singular value decomposition (SVD) of rectangular matrices. It is fast (somewhat faster than QR and somewhat slower than divide and conquer (DC); see [6]) and is accurate in a relative sense (which cannot be said for QR and DC; see [1]). It also requires less workspace and applies to a wider class of matrices (the matrix elements can vary in a wider range of magnitudes) than its competitors. The modified Jacobi method first prepares the initial matrix for the iteration by applying to it one or two QR factorizations (cf. [19, 4, 5]). In the iterative part, many ideas from [3] have been implemented. In addition, to better exploit the computer resources, like the cache memory, it uses a new block-oriented, quasi-cyclic pivot strategy. This strategy mimics the BLAS blocking, thus exploiting contiguous memory benefits. It also takes into account the fact that after QR factorization, the larger elements typically lie in the vicinity of the diagonal, so more work has to be done within the diagonal blocks.

Still lacking are the global and the asymptotic convergence proofs of the modification of [5, 6]. The latter is less critical, since it is well known that Jacobi methods are quadratically convergent per sweep under usual cyclic strategies (see [20, 11]), and therefore, the quasi-cyclic methods should be (at least; see [17]) quadratically convergent per quasi sweep.

The aim of this paper is to fill the missing gap, i.e., to prove the global convergence of the Jacobi method, defined by the pivot strategy proposed in [6]. The proof presented here covers the main algorithm from [6, Algorithm 1, case $k = 1$]. The other cases ($k = 0$, which is trivial, and $k = 2$, which is even more complicated) can be proved by the same technique.

Note that the global convergence of a one-sided SVD Jacobi method for the matrix $G$ actually means the global convergence of the corresponding two-sided Jacobi method for the nonnegative definite Gram matrix $G^T G$. For this reason, we consider here the two-sided Jacobi method for general symmetric matrices, under the special quasi-cyclic pivot strategy which has been used in [6].

For any quadratic matrix $X = (x_{ij})$, the function $\mathrm{Off}(X) = \|X - \mathrm{diag}(X)\|_F$, where $\mathrm{diag}(X)$ is the diagonal part of $X$ and $\|\cdot\|_F$ is the Frobenius norm, is referred

---

[*] Received by the editors November 17, 2005; accepted for publication (in revised form) by N. J. Higham August 30, 2006; published electronically March 19, 2007.
http://www.siam.org/journals/simax/29-2/64552.html

[†] Department of Mathematics, University of Zagreb, 10000 Zagreb, Croatia (hari@math.hr).

to as [illegible] of $X$. Then, for any symmetric $A$, a useful measure of almost diagonality is

$$S(A) = \mathrm{Off}(A)/\sqrt{2} = \sqrt{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} |a_{ij}|^2}.$$

In [15] Mascarenhas has proved that the diagonal elements of the iterated symmetric matrix $A^{(k)}$, obtained by the Jacobi method under any pivot strategy, converge. Therefore, it is sufficient to prove that for any initial symmetric matrix $A$, $S(A^{(k)}) \to 0$ as $k \to \infty$. Actually, we shall prove that under the pivot strategy used in [6],

$$S^2(A^{(M)}) \le t_n S^2(A), \quad 0 \le t_n < 1. \tag{1.1}$$

Here $A^{(M)}$ is obtained from $A$ after one quasi sweep, and $t_n$ depends just on $n$. This is sufficient for the proof since the sequence $S(A^{(k)})$ is nonincreasing.

The paper is divided into three sections. In section 2 we introduce notation and the quasi-cyclic strategy, and we formulate the main theorem. In section 3 we prove the theorem.

**2. The quasi-cyclic Jacobi method $J_{\mathcal{M}}$.** Although the method and the proof can be considered for complex Hermitian matrices, for simplicity we restrict our considerations to real symmetric matrices.

**2.1. Simple symmetric quasi-cyclic Jacobi methods.** A (two-sided) Jacobi method for diagonalizing a symmetric $n \times n$ matrix $A$ performs a sequence of similarity transformations

$$A^{(k+1)} = [R^{(k)}]^T A^{(k)} R^{(k)}, \quad V^{(k+1)} = V^{(k)} R^{(k)}, \quad k \ge 0\,, \tag{2.1}$$

where $A^{(0)} = A$, $V^{(0)} = I_n$, and $R^{(0)}, R^{(1)}, \dots$ are plane rotations. For each $k$, $R^{(k)}$ is defined by a pair of indices $(p,q) = (p(k), q(k))$ called a [illegible] and by four essential elements $R_{pp}^{(k)} = R_{qq}^{(k)} = \cos\phi_k$, $R_{pq}^{(k)} = -R_{qp}^{(k)} = \sin\phi_k$. All other elements of $R^{(k)}$ are as in the identity matrix $I_n$. The process (2.1) is defined by a rule for computing the elements of $R^{(k)}$ and by a way of selecting pivot pairs ([illegible]). We assume the standard angle choice

$$\tan 2\phi_k = \frac{2a_{pq}^{(k)}}{a_{qq}^{(k)} - a_{pp}^{(k)}}, \quad \phi_k \in \left[\frac{-\pi}{4}, \frac{\pi}{4}\right]\,, \tag{2.2}$$

which makes the [illegible] $a_{pq}^{(k)}$ zero; i.e., $a_{p(k)q(k)}^{(k+1)} = 0$, $k \ge 0$, holds. Consequently, we have

$$S^2(A^{(k+1)}) = S^2(A^{(k)}) \; - \; \left(a_{p(k)q(k)}^{(k)}\right)^2, \quad k \ge 0,$$

and $\lim_{k\to\infty} a_{p(k)q(k)}^{(k)} = 0$. Here we have assumed $A^{(k)} = (a_{ij}^{(k)})$.

If $A$ is a $1 \times 1$ matrix, we set $S(A) = 0$, and if $A$ is $2 \times 2$, the process is completed for $k = 1$ since $S(A^{(1)}) = 0$. So, we assume $n \ge 3$.

Let $N = n(n-1)/2$, $\mathbf{P}_n = \{(i,j) : 1 \le i < j \le n\}$ and $\mathbf{N}_0 = \{0, 1, 2, \dots\}$. Each pivot strategy can be identified with a function $\mathcal{I}$ from $\mathbf{N}_0$ to $\mathbf{P}_n$, defined by $\mathcal{I}(k) = (p(k), q(k))$, $k \ge 0$. If $\mathcal{I}$ is periodic, then $\mathcal{I}$ is called a [illegible]. Let $\mathcal{I}$

be a periodic strategy with period $M$. If $\{\mathcal{I}(k) : 0 \le k \le M-1\} = \mathbf{P}_n$ and $M > N$ ($M = N$), then $\mathcal{I}$ is called a *quasi-cyclic* (*cyclic*) *strategy*.

Let $\mathbf{S}$ be a subset of $\mathbf{P}_n$ and let $\nu(\mathbf{S})$ denote its cardinality. By $\mathbf{O}(\mathbf{S})$ we mean a collection of all finite sequences made of the elements from $\mathbf{S}$. If $O \in \mathbf{O}(\mathbf{S})$, we assume that each element of $\mathbf{S}$ appears at least once in $O$ (otherwise $\mathbf{S}$ is replaced with some of its proper subsets). Thus, each sequence from $\mathbf{O}(\mathbf{S})$ contains at least $\nu(\mathbf{S})$ terms. If $\mathbf{S}$ is an empty set, $\nu(\mathbf{S}) = 0$ and $\mathbf{O}(\mathbf{S})$ consists of a singleton, which is an empty sequence.

A cyclic or a quasi-cyclic strategy can be specified in the following way. For any $O = \{(i_r, j_r)\}_{r=0}^{M-1} \in \mathbf{O}(\mathbf{P}_n)$, the cyclic or the quasi-cyclic strategy $\mathcal{I}_O$, generated by $O$, is given by

$$\mathcal{I}_O(k) \equiv (p(k), q(k)) = (i_r, j_r), \quad 0 \le r \le M-1, \quad k \ge 0,$$

provided that $k \equiv r \pmod{M}$. Thus,

$$\begin{aligned}
&(p(0), q(0)) = (i_0, j_0), \ (p(1), q(1)) = (i_1, j_1), \ldots, \\
&(p(M-1), q(M-1)) = (i_{M-1}, j_{M-1}), \ (p(M), q(M)) = (i_0, j_0), \\
&(p(M+1), q(M+1)) = (i_1, j_1), \ldots .
\end{aligned}$$

We denote the Jacobi method, which defines $\phi_k$ by (2.2) and uses the pivot strategy $\mathcal{I}$ by $J(\mathcal{I})$. $J(\mathcal{I})$ is called a quasi-cyclic/cyclic (Jacobi) method if $\mathcal{I}$ is the quasi-cyclic/cyclic strategy. Since $n$ is fixed, in what follows we write $\mathbf{P}$ for $\mathbf{P}_n$.

Let $\mathbf{S}$ be any subset of $\mathbf{P}$. By $O_R(\mathbf{S})$ we denote the "rowwise ordering" of $\mathbf{S}$, that is, the sequence satisfying the following two conditions: (i) Each element $(i,j) \in \mathbf{S}$ appears exactly once in $O_R(\mathbf{S})$; (ii) for any two terms $(i_1, j_1)$ and $(i_2, j_2)$ in $O_R(\mathbf{S})$, $(i_1, j_1)$ precedes $(i_2, j_2)$ if $i_1 < i_2$ or $i_1 = i_2$ and $j_1 < j_2$. In an obvious manner we can define the "columnwise" ordering of $\mathbf{S}$ denoted by $O_C(\mathbf{S})$.

Let $\mathbf{S}_i$, $1 \le i \le \sigma$, be subsets of $\mathbf{P}$ and let $O_i \in \mathbf{O}(\mathbf{S}_i)$, $1 \le i \le \sigma$, be arbitrary sequences. By $[O_1, O_2, \ldots, O_\sigma]$, we mean the sequence which is obtained by the concatenation of the sequences $O_1, O_2, \ldots, O_\sigma$. Often, we shall omit the brackets.

**2.2. Quasi-cyclic strategies.** To prove the convergence theorem (Theorem 2.1 below), we need the notion of equivalent strategies. This notion is extended from cyclic [8, 9, 18] to quasi-cyclic strategies, as it has been done in [17]. In what follows, *strategy* means *quasi-cyclic* or *cyclic strategy*.

Let $\mathbf{S}$ be a subset of $\mathbf{P}$. Let $O = \{(i_r, j_r)\}_{r=0}^{s} \in \mathbf{O}(\mathbf{S})$. An admissible transposition on $O$ is any transposition of two adjacent terms $(i_r, j_r), (i_{r+1}, j_{r+1}) \to (i_{r+1}, j_{r+1}), (i_r, j_r)$, provided that the sets $\{i_r, j_r\}$ and $\{i_{r+1}, j_{r+1}\}$ are disjoint. Such pairs will also be called *commuting* or *disjoint*.

The sequences $O, O' \in \mathbf{O}(\mathbf{S})$ are *equivalent* if one can be obtained from the other by a finite number of admissible transpositions. In this case we write $O \sim O'$. For example, if $\mathbf{S} = \{(1,2), (2,3), (1,4)\}$, then $O = (1,2), (1,2), (1,4), (2,3), (2,3) \in \mathbf{O}(\mathbf{S})$ and $O' = (1,2), (1,2), (2,3), (2,3), (1,4) \in \mathbf{O}(\mathbf{S})$ are equivalent.

Let $\mathcal{I}$ be a strategy with period $M$. By $O_{\mathcal{I}}$ we mean the sequence $\{\mathcal{I}(k)\}_{k=0}^{M-1}$. If $\mathcal{I}$ and $\mathcal{I}'$ are two strategies with the same period $M$, then $\mathcal{I}$ and $\mathcal{I}'$ are equivalent if $O_{\mathcal{I}} \sim O_{\mathcal{I}'}$. In such a case we write $\mathcal{I} \sim \mathcal{I}'$.

For a given symmetric matrix $A$ and a given strategy $\mathcal{I}$, the sequence $\{R^{(k)}\}$ is well defined. Let $T_{\mathcal{I}(k)}(X)$ mean $[R^{(k)}]^T X R^{(k)}$. Then the recursion in (2.1) can be written $A^{(k+1)} = T_{\mathcal{I}(k)}(A^{(k)})$, implying

$$A^{(k+1)} = T_{\mathcal{I}(k)} T_{\mathcal{I}(k-1)} \cdots T_{\mathcal{I}(0)}(A), \quad k \ge 0,$$

where $T_{\mathcal{I}(k)}T_{\mathcal{I}(k-1)}\cdots T_{\mathcal{I}(0)}$ denotes the composition of the underlying linear operators. Let

$$T_k^{\mathcal{I}} = T_{\mathcal{I}(k-1)}T_{\mathcal{I}(k-2)}\cdots T_{\mathcal{I}(0)}, \quad k \geq 1; \qquad T_0^{\mathcal{I}} = E,$$

where $E$ is the identity operator. Note that $T_{\mathcal{I}(k)}$ commutes with $T_{\mathcal{I}(k+1)}$ if $(p(k), q(k))$ and $(p(k+1),\ q(k+1))$ are commuting. Hence using the same argument as in [11, Lemma 1.1], we obtain the following implication (see also [8]):

$$\text{If } \mathcal{I} \sim \mathcal{I}', \text{ then } T_{kM}^{\mathcal{I}}(A) = T_{kM}^{\mathcal{I}'}(A), \ k \geq 0. \tag{2.3}$$

That is, equivalent strategies give the same matrices at the end of each quasi sweep consisting of $M$ iterations of the form (2.1). Since $\{S(A^{(k)})\}_{k=0}^{\infty}$ makes a nonincreasing sequence, (2.3) implies that $J(\mathcal{I})$ is convergent iff $J(\mathcal{I}')$ is convergent, provided that $\mathcal{I} \sim \mathcal{I}'$. In particular, if the relation (1.1) holds for $\mathcal{I}'$ and $\mathcal{I} \sim \mathcal{I}'$, then it holds for $\mathcal{I}$.

**2.3. The special quasi-cyclic block-oriented method.** Let $A = (a_{rs})$ be a real symmetric matrix of order $n$. The starting point in describing the method is the following block-partition of $A$:

$$A = \left[\begin{array}{cccc} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mm} \end{array}\right] \begin{array}{l} \}\, n_1 \\ \}\, n_2 \\ \vdots \\ \}\, n_m \end{array},$$

where $A_{ij}$ is an $n_i \times n_j$ block of $A$. Since $n_1 + \cdots + n_m = n$, $\mathcal{M} = (n_1, n_2, \ldots, n_m)$ is a partition of $n$. In practical computations, it is desirable that no $n_i$ exceeds some $n_0$ which depends on the machine and its available fast (cache) memory. The cache memory is several times faster than the main computer memory (according to [2, section 1.3.2] 5 to 10 times faster).

With each submatrix $B$ contained in the uppertriangle of $A$ (or with each principal submatrix $B$ of $A$) one can associate the set $\mathbf{S}(B) \subseteq \mathbf{P}$ of those pairs which are subscripts of the elements of $B$ (and of the upper triangle of $B$). By abuse of notation, the row- and columnwise orderings of $\mathbf{S}(B)$ are denoted by $O_R(B)$ $(= O_R(\mathbf{S}(B)))$ and $O_C(B)$ $(= O_C(\mathbf{S}(B)))$, respectively. We shall also use $\mathbf{S}_{ij} = \mathbf{S}(A_{ij})$ and $\mathcal{R}_{ij} = O_R(A_{ij})$ $(= O_R(\mathbf{S}_{ij}))$, $\mathcal{C}_{ij} = O_C(A_{ij})$ $(= O_C(\mathbf{S}_{ij}))$ for $i \leq j$.

Let $\mathcal{O}_{\mathcal{M}}$ denote the sequence of pairs which defines one quasi sweep of the method from [6]. It is defined as follows.

First, define sequences

$$\mathcal{R}_i = [\mathcal{R}_{i+1,i+1}, \mathcal{R}_{ii}, \mathcal{R}_{i,i+1}, \mathcal{R}_{i,i+2}, \ldots, \mathcal{R}_{im}], \quad 1 \leq i \leq m-1.$$

Then,

$$\mathcal{O}_{\mathcal{M}} = [\mathcal{R}_{11}, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{m-1}, \mathcal{R}_{mm}]. \tag{2.4}$$

The quasi-cyclic Jacobi method from [6] is defined by the strategy $\mathcal{I}_{\mathcal{M}} = \mathcal{I}_{\mathcal{O}_{\mathcal{M}}}$ and we denote it by $J_{\mathcal{M}}$. Its pivot strategy $\mathcal{I}_{\mathcal{M}}$ uses the rowwise ordering within each block $A_{ij}$. $\mathcal{I}_{\mathcal{M}}$ fetches the blocks in a rowwise fashion, and all the diagonal blocks $A_{ii}$ are operated twice. Therefore, we call it the [illegible]. The quasi sweep contains $M$ ordinary Jacobi steps, where $M$ is the number of pairs in the sequence $\mathcal{O}_{\mathcal{M}}$ from (2.4). The aim of this paper is to prove the global convergence of $J_{\mathcal{M}}$. In particular, we shall prove the following.

THEOREM 2.1. [illegible] $m \geq 1$ [illegible] $n \geq \max\{m, 2\}$ [illegible]

[illegible] $\mathcal{M} = (n_1, \ldots, n_m)$ [illegible] $n$ [illegible] $m$ [illegible] $A$ [illegible] $n$ [illegible] $(A_{ij})$ [illegible] $A$ [illegible] $m$ [illegible] $m$ [illegible] $A_{ii}$ [illegible] $n_i \times n_i$ $1 \leq i \leq m$ [illegible] $J_{\mathcal{M}}$ [illegible] $A$ [illegible] $A^{(0)} = A$, $A^{(1)}$, $A^{(2)}, \ldots$ [illegible]

$$M = n(n-1)/2 + \sum_{i=1}^{m} n_i(n_i - 1)/2$$

[illegible]

$$S^2(A^{(M)}) \leq t_n^{(\mathcal{M})}\, S^2(A), \quad 0 \leq t_n^{(\mathcal{M})} < 1, \tag{2.5}$$

[illegible] $\mathcal{M}$ [illegible] $t_n^{(\mathcal{M})}$ [illegible] $n$

**3. Proof of Theorem 2.1.** To prove Theorem 2.1, it is sufficient to prove (2.5) for some strategy $\mathcal{I}$ which is equivalent to $\mathcal{I}_{\mathcal{M}}$. We shall find such an $\mathcal{I}$ with the special property that it fetches the matrix elements by block-columns, and within each block-column by columns, plus some extra work within the diagonal blocks.

**3.1. Obtaining $\mathcal{I}$.** We call two sets $\mathbf{S}_1$ and $\mathbf{S}_2$ from $\mathbf{P}$ commuting if each pair from $\mathbf{S}_1$ commutes with all the pairs from $\mathbf{S}_2$. In the same way we define two commuting sequences made of the elements from $\mathbf{P}$. Obviously, if $\mathbf{S}_1$ and $\mathbf{S}_2$ are commuting, then each sequence made of the elements from $\mathbf{S}_1$ commutes with all the sequences made of the elements from $\mathbf{S}_2$ and vice versa. If the two commuting subsequences take adjacent positions in the sequence which defines some pivot strategy, then we can interchange the positions of these two subsequences, obtaining an equivalent (sequence and) pivot strategy.

To define $\mathcal{I}$, we start with the sequence $\mathcal{O}_{\mathcal{M}}$ and transform it using admissible transpositions. We divide the process into several stages. In the first stage, we transform $\mathcal{O}_{\mathcal{M}}$ into $\mathcal{O}_1$; in the second stage we transform $\mathcal{O}_1$ into $\mathcal{O}_2$; and finally, we transform $\mathcal{O}_2$ into $\mathcal{O}$ which defines $\mathcal{I}$.

We shall illustrate the transitions for the case $m = 5$. After this illustration, we shall provide the result which holds for general $m$. Except for the first line, we shall omit writing [ ] which denotes the concatenation of sequences. Vertical bars | are used only to better recognize the obtained subsequences. We have

$$\begin{aligned}\mathcal{O}_{\mathcal{M}} &= [\mathcal{R}_{11}, \mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4, \mathcal{R}_{55}] \\ &= \mathcal{R}_{11}, |\, \mathcal{R}_{22}, \mathcal{R}_{11}, \mathcal{R}_{12}, \mathcal{R}_{13}, \mathcal{R}_{14}, \mathcal{R}_{15}, |\, \mathcal{R}_{33}, \mathcal{R}_{22}, \mathcal{R}_{23}, \mathcal{R}_{24}, \mathcal{R}_{25}, | \\ &\quad\ \mathcal{R}_{44}, \mathcal{R}_{33}, \mathcal{R}_{34}, \mathcal{R}_{35} |\, \mathcal{R}_{55}, \mathcal{R}_{44}, \mathcal{R}_{45}, |\, \mathcal{R}_{55}\,.\end{aligned}$$

Note that $\mathcal{R}_{ii}$ commutes with $\mathcal{R}_{i+1,i+1}$ and that $\mathcal{R}_{ij}$ commutes with $\mathcal{R}_{pq}$ provided that $(i, j)$ and $(p, q)$ are commuting. Using these facts, we can "move left" the subsequence $\mathcal{R}_{22}$ from the ninth position to the fifth position, just behind $\mathcal{R}_{12}$, and we can "move right" $\mathcal{R}_{22}$ from the second position to the third position. Continuing this process, we obtain the following equivalent sequences:

$$\begin{aligned}\mathcal{O}_{\mathcal{M}} \sim\ & \mathcal{R}_{11}, | \,\mathcal{R}_{11}, | \,\mathcal{R}_{22}, \mathcal{R}_{12}, \mathcal{R}_{22}, | \,\mathcal{R}_{13}, \mathcal{R}_{14}, \mathcal{R}_{15}, \mathcal{R}_{33}, \mathcal{R}_{23}, \mathcal{R}_{24}, \mathcal{R}_{25},\\ & \quad \mathcal{R}_{44}, \mathcal{R}_{33}, \mathcal{R}_{34}, \mathcal{R}_{35}, | \,\mathcal{R}_{55}, \mathcal{R}_{44}, \mathcal{R}_{45}, | \,\mathcal{R}_{55}\\ \sim\ & \mathcal{R}_{11},\ \mathcal{R}_{11}, | \,\mathcal{R}_{22}, \mathcal{R}_{12}, \mathcal{R}_{22}, | \,\mathcal{R}_{13}, \mathcal{R}_{33}, \mathcal{R}_{23}, \mathcal{R}_{33}, | \,\mathcal{R}_{14}, \mathcal{R}_{15}, \mathcal{R}_{24}, \mathcal{R}_{25},\\ & \mathcal{R}_{44}, \mathcal{R}_{34}, \mathcal{R}_{35}, \mathcal{R}_{55}, \mathcal{R}_{44}, \mathcal{R}_{45}, | \,\mathcal{R}_{55}\\ \sim\ & \mathcal{R}_{11},\ \mathcal{R}_{11}, | \,\mathcal{R}_{22}, \mathcal{R}_{12}, \mathcal{R}_{22}, | \,\mathcal{R}_{13}, \mathcal{R}_{33}, \mathcal{R}_{23}, \mathcal{R}_{33}, | \,\mathcal{R}_{14}, \mathcal{R}_{24}, \mathcal{R}_{44}, \mathcal{R}_{34}, |\\ & \quad \mathcal{R}_{15}, \mathcal{R}_{25}, \mathcal{R}_{35}, \mathcal{R}_{55}, \mathcal{R}_{44}, \mathcal{R}_{45}, | \,\mathcal{R}_{55}\\ \sim\ & \mathcal{R}_{11},\ \mathcal{R}_{11}, | \,\mathcal{R}_{22}, \mathcal{R}_{12}, \mathcal{R}_{22}, | \,\mathcal{R}_{13}, \mathcal{R}_{33}, \mathcal{R}_{23}, \mathcal{R}_{33}, |\\ & \quad \mathcal{R}_{14}, \mathcal{R}_{24}, \mathcal{R}_{44}, \mathcal{R}_{34}, \mathcal{R}_{44}, | \,\mathcal{R}_{15}, \mathcal{R}_{25}, \mathcal{R}_{35}, \mathcal{R}_{55}, \mathcal{R}_{45},\ \mathcal{R}_{55}\,.\end{aligned}$$

Using the same technique, we can conclude that for general $m$, $\mathcal{O}_{\mathcal{M}} \sim \mathcal{O}_1$, where

$$\begin{aligned}\mathcal{O}_1 = \mathcal{R}_{11},\ & \mathcal{R}_{11}, | \,\mathcal{R}_{22}, \mathcal{R}_{12}, \mathcal{R}_{22}, | \,\mathcal{R}_{13}, \mathcal{R}_{33}, \mathcal{R}_{23}, \mathcal{R}_{33}, | \dots\\ & | \,\mathcal{R}_{1m}, \mathcal{R}_{2m}, \mathcal{R}_{3m}, \dots, \mathcal{R}_{m-2,m}, \mathcal{R}_{mm}, \mathcal{R}_{m-1,m}, \mathcal{R}_{mm}.\end{aligned}$$

We see that $\mathcal{O}_1$ is obtained by the concatenation of $m$ subsequences, the $j$th one having the form

$$\mathcal{R}_{1j}, \mathcal{R}_{2j}, \mathcal{R}_{3j}, \dots, \mathcal{R}_{j-2,j}, \mathcal{R}_{jj}, \mathcal{R}_{j-1,j}\mathcal{R}_{jj}, \quad 3 \le j \le m.$$

Note that the first subsequence $\mathcal{R}_{11}, \mathcal{R}_{11}$ and the second one $\mathcal{R}_{22}, \mathcal{R}_{12}, \mathcal{R}_{22}$ also fit into this pattern. Therefore, we can write

$$\mathcal{O}_{\mathcal{M}} \sim \square_{j=1}^{m}\ \mathcal{R}_{1j}, \mathcal{R}_{2j}, \mathcal{R}_{3j}, \dots, \mathcal{R}_{j-2,j}, \mathcal{R}_{j,j}, \mathcal{R}_{j-1,j}\mathcal{R}_{j,j}, \tag{3.1}$$

where $\square$ stands for the concatenation. Now, remember that (see [8, 9]) the row-cyclic and the column-cyclic orderings are equivalent. This means that we can replace each $\mathcal{R}_{ii}$ with $\mathcal{C}_{ii}$. Furthermore, the rowwise ordering of each rectangular block is equivalent to the columnwise ordering of that block. Hence, for $3 \le j \le m$, we have

$$\mathcal{R}_{1j}, \mathcal{R}_{2j}, \mathcal{R}_{3j}, \dots, \mathcal{R}_{j-2,j} \sim O_R(B_j) \sim O_C(B_j) \equiv \mathsf{C}_j\,, \text{ where } B_j = \begin{bmatrix} A_{1j} \\ A_{2j} \\ \vdots \\ A_{j-2,j} \end{bmatrix}.$$

Note that $B_j \in \mathbf{R}^{s_{j-2} \times n_j}$, where

$$s_j = n_1 + n_2 + \cdots + n_j, \quad 1 \le j \le m, \quad s_0 = 0.$$

Let

$$\mathsf{c}_{jk} = O_C(B_j e_k) = (1, s_{j-1}+k), (2, s_{j-1}+k), \dots, (s_{j-2}, s_{j-1}+k), \quad 1 \le k \le n_j,$$

where $e_k$ is the $k$th column of the identity matrix, so that $B_j e_k$ is the $k$th column of $B_j$. Then

$$\mathsf{C}_j = \mathsf{c}_{j1}, \mathsf{c}_{j2}, \dots, \mathsf{c}_{jn_j}, \quad 3 \le j \le m\,.$$

Furthermore, we can write

$$\mathcal{R}_{1j}, \mathcal{R}_{2j}, \mathcal{R}_{3j}, \dots, \mathcal{R}_{j-2,j}, \mathcal{R}_{jj}, \mathcal{R}_{j-1,j}\mathcal{R}_{jj} \sim \mathsf{C}_j, \mathcal{C}_{jj}, \mathcal{C}_{j-1,j}, \mathcal{C}_{jj} \equiv \mathbf{C}_j\,. \tag{3.2}$$

If we set

$$\begin{aligned} \mathsf{c}'_{jk} &= O_C(A_{j-1,j}e_k), \quad 1 \le k \le n_j, \quad \text{and} \\ \mathsf{c}''_{jk} &= O_C([e_1,\dots,e_{k-1}]^T A_{jj}e_k), \quad 2 \le k \le n_j, \end{aligned}$$

then

$$\mathcal{C}_{j-1,j} = \mathsf{c}'_{j1},\mathsf{c}'_{j2},\dots,\mathsf{c}'_{jn_j} \quad \text{and} \quad \mathcal{C}_{jj} = \mathsf{c}''_{j2},\mathsf{c}''_{j3},\dots,\mathsf{c}''_{jn_j}.$$

Hence we can write for $3 \le j \le m$,

$$\mathbf{C}_j = \mathsf{c}_{j1},\mathsf{c}_{j2},\dots,\mathsf{c}_{jn_j},\mathsf{c}''_{j2},\dots,\mathsf{c}''_{jn_j},\mathsf{c}'_{j1},\mathsf{c}'_{j2},\dots,\mathsf{c}'_{jn_j},\mathsf{c}''_{j2},\dots,\mathsf{c}''_{jn_j}. \tag{3.3}$$

For $j = 1$ the above formula reduces to $\mathsf{c}''_{12},\dots,\mathsf{c}''_{1n_1},\mathsf{c}''_{12},\dots,\mathsf{c}''_{1n_1}$ and for $j = 2$ to $\mathsf{c}''_{22},\dots,\mathsf{c}''_{2n_2},\mathsf{c}'_{21},\mathsf{c}'_{22},\dots,\mathsf{c}'_{2n_2},\mathsf{c}''_{22},\dots,\mathsf{c}''_{2n_2}$.

LEMMA 3.1. [illegible]

(i) $\mathbf{C}_j \sim \mathcal{C}_j = \mathsf{c}_{j1},\mathsf{c}_{j2},\mathsf{c}''_{j2},\mathsf{c}_{j3},\mathsf{c}''_{j3},\dots,\mathsf{c}_{jn_j},\mathsf{c}''_{jn_j},\mathsf{c}'_{j1},\mathsf{c}'_{j2},\mathsf{c}''_{j2},\dots,\mathsf{c}'_{jn_j},\mathsf{c}''_{jn_j}$ $3 \le j \le m$

(ii) [illegible] $\mathcal{C}_1 = \mathcal{C}_{11},\mathcal{C}_{11}$ $\mathcal{C}_2 = \mathcal{C}_{22},\mathsf{c}'_{21},\mathsf{c}'_{22},\mathsf{c}''_{22},\dots,\mathsf{c}'_{2n_2},\mathsf{c}''_{2n_2}$ [illegible]

$$\mathcal{O}_{\mathcal{M}} \sim \mathcal{O} = \Box_{j=1}^{m} \mathcal{C}_j\,. \tag{3.4}$$

[illegible] (i) We start from the sequence $\mathbf{C}_j$ in the relation (3.3) and transform it by admissible transformations into $\mathcal{C}_j$. Since $\mathsf{c}''_{j2}$ commutes with the subsequence $\mathsf{c}_{j3},\dots,\mathsf{c}_{jn_j}$, we can move it left to the position just behind $\mathsf{c}_{j2}$. In a similar fashion, we move $\mathsf{c}''_{j3}$ just behind $\mathsf{c}_{j3}$ and so on. Note that, $\mathsf{c}'_{j1},\dots,\mathsf{c}'_{jn_j},\mathsf{c}''_{j2},\dots,\mathsf{c}''_{jn_j}$, can be replaced with $\mathsf{c}'_{j1},\mathsf{c}'_{j2},\mathsf{c}''_{j2},\dots,\mathsf{c}'_{jn_j},\mathsf{c}''_{jn_j}$, which is the columnwise ordering of $[A^T_{j-1,j},A^T_{jj}]^T$.

(ii) As has been shown in the lines after the relation (3.3),

$$\begin{aligned} \mathcal{R}_{11}\mathcal{R}_{11} &\sim \mathbf{C}_1 = \mathcal{C}_{11}\mathcal{C}_{11} = \mathcal{C}_1, \\ \mathcal{R}_{22}\mathcal{R}_{12}\mathcal{R}_{22} &\sim \mathbf{C}_2 = \mathcal{C}_{22}\mathcal{C}_{12}\mathcal{C}_{22} \sim \mathcal{C}_2\,. \end{aligned}$$

Using (3.1), (3.2), (3.3), and (i), we obtain (3.4). □

Thus, in considering the off-norm reduction after one quasi sweep of the original method $J_{\mathcal{M}}$, we can replace the initial pivot strategy defined by $\mathcal{O}_{\mathcal{M}}$ with the pivot strategy defined by $\mathcal{O}$.

To prove Theorem 2.1, we use mathematical induction with respect to $m$.

**The induction basis.** For $m = 1$, $\mathcal{O}_{\mathcal{M}} = \mathcal{R}_{11}\mathcal{R}_{11} \sim \mathcal{C}_{11}\mathcal{C}_{11} = \mathcal{O}$ and by the well-known result of Henrici and Zimmermann [14], we have

$$S^2(A^{(M)}) \le \left(1 - 2^{-\frac{(n_1-1)(n_1-2)}{2}}\right)^2 S^2(A)\,.$$

Since for $m = 1$, $n = n_1$, we can take $t_n^{(\mathcal{M})} = (1 - 2^{-\frac{n^2}{2}})^2$, which is smaller than 1 for all $n \ge 2$.

For $m = 2$, $\mathcal{O}_{\mathcal{M}} = \mathcal{R}_{11}\mathcal{R}_{22}\mathcal{R}_{11}\mathcal{R}_{12}\mathcal{R}_{22} \sim \mathcal{O}$, and hence by the same result,

$$S^2(A^{(M)}) \le \left(1 - 2^{-\frac{(n-1)(n-2)}{2}}\right) S^2(A^{(\frac{n_1(n_1-1)}{2}+\frac{n_2(n_2-1)}{2})}) \le \left(1 - 2^{-\frac{(n-1)(n-2)}{2}}\right) S^2(A)\,,$$

so we can take $t_n^{(\mathcal{M})} = 1 - 2^{-\frac{n^2}{2}}$, $n \ge 2$.

**The induction hypothesis.** Let us assume that the assertion of Theorem 2.1 holds for $m' = m-1$. This means that (2.5) holds for any partition $\mathcal{M}' = (n'_1, \ldots, n'_{m'})$ of $n'$, $n' \geq \max\{2, m'\}$, and for any symmetric matrix $A'$ of order $n'$ such that

(a) $A'$ is partitioned into $m'$ block-rows and block-columns, according to the partition $\mathcal{M}'$.

(b) the sequence $A^{(0)} = A'$, $A^{(1)}$,… is generated by the Jacobi method, which is defined by the quasi-cyclic strategy $\mathcal{I}_{\mathcal{O}_{\mathcal{M}'}}$, and $\mathcal{O}_{\mathcal{M}'}$ is defined by (2.4), provided that $m$ and $\mathcal{M}$ are replaced with $m'$ and $\mathcal{M}'$, respectively.

Since $\mathcal{O}_{\mathcal{M}'} \sim \mathcal{O}'$, where $\mathcal{O}'$ is defined as in (3.4), the induction hypothesis implies that (2.5) holds for the strategy defined by $\mathcal{O}'$.

**3.2. The induction step.** To prove the induction step, we assume that $\mathcal{M} = (n_1, \ldots, n_m)$ is an arbitrary partition of $n$, $n \geq m \geq 3$, and $A = (A_{ij})$ is an arbitrary symmetric matrix of order $n$, partitioned into $m$ block-rows and $m$ block-columns, according to the partition $\mathcal{M}$. We have to prove Theorem 2.1 for $A$ and $J_{\mathcal{M}}$ defined by $\mathcal{O}_{\mathcal{M}}$, provided the induction hypothesis holds.

Since $\mathcal{O}_{\mathcal{M}} \sim \mathcal{O}$, the same matrix $A^{(M)}$ is obtained if the quasi-cyclic Jacobi method $J_{\mathcal{M}}$ is replaced with $J$, which is defined by $\mathcal{O}$. So, from here on, we assume that $A = A^{(0)}$, $A^{(1)}$,…,$A^{(M)}$ are generated by $J$.

**Notation.** Let $I_n = [e_1, \ldots, e_n]$ be the column-partition of the identity matrix. Using the coordinate vectors $e_j$, we define

$$E_j = [e_1, \ldots, e_j], \ 1 \leq j \leq n,$$

so that $E_n = I_n$. Let

$$\begin{aligned} \tilde{n} &= s_{m-1} = n_1 + n_2 + \cdots + n_{m-1}, \\ \tilde{M} &= \frac{\tilde{n}(\tilde{n}-1)}{2} + \sum_{i=1}^{m-1} \frac{n_i(n_i-1)}{2}, \\ \tilde{A} &= A^{(\tilde{M})}, \quad B = E_{\tilde{n}}^T A E_{\tilde{n}}, \quad \tilde{B} = E_{\tilde{n}}^T \tilde{A} E_{\tilde{n}}, \\ \Sigma^2 &= S^2(A)\,, \quad S^2(B) = \varepsilon\Sigma^2\,, \quad 0 \leq \varepsilon \leq 1\,. \end{aligned}$$

Note that after $\tilde{M}$ steps, one full quasi sweep is performed on $B$. The underlying strategy is defined by $\tilde{\mathcal{O}}$, which is defined via $\tilde{\mathcal{M}} = (n_1, \ldots, n_{m-1})$. This $\tilde{\mathcal{O}}$ is one of those $\mathcal{O}'$ described above, and the induction hypothesis can be applied to $B$. We obtain

$$S^2(\tilde{B}) \leq t_{\tilde{n}}^{(\tilde{\mathcal{M}})} S^2(B) = t_{\tilde{n}}^{(\tilde{\mathcal{M}})} \varepsilon \Sigma^2, \quad 0 \leq t_{\tilde{n}}^{(\tilde{\mathcal{M}})} < 1\,.$$

Hence

$$S^2(\tilde{A}) = S^2(\tilde{B}) + (1-\varepsilon)\Sigma^2 \leq t_{\tilde{n}}^{(\tilde{\mathcal{M}})} \varepsilon \Sigma^2 + (1-\varepsilon)\Sigma^2 = [1 - \varepsilon(1 - t_{\tilde{n}}^{(\tilde{\mathcal{M}})})]\Sigma^2\,.$$

Although $S^2(A^{(M)}) \leq S^2(\tilde{A})$, we cannot set $t_n^{(\mathcal{M})} = 1 - \varepsilon(1 - t_{\tilde{n}}^{(\tilde{\mathcal{M}})})$, since $\varepsilon$ can be arbitrarily small or even zero. Therefore, we have to estimate the contribution to the further off-norm reduction coming from the last $M - \tilde{M}$ steps.

Let

$$\begin{aligned} \varepsilon_1 \Sigma^2 &= S^2(E_{\tilde{n}-n_{m-1}}^T \tilde{A} E_{\tilde{n}-n_{m-1}}), \\ \varepsilon_2 \Sigma^2 &= \textstyle\sum_{i=1}^{m-2} \|\tilde{A}_{i,m-1}\|_F^2, \\ \varepsilon_3 \Sigma^2 &= S^2(\tilde{A}_{m-1,m-1})\,. \end{aligned} \tag{3.5}$$

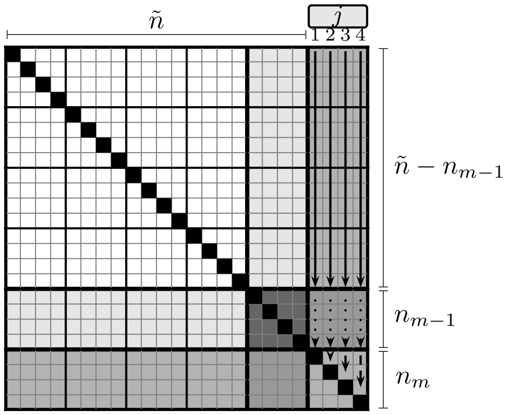


FIG. 1. *Matrices $\tilde{A}$ and $\tilde{B}$ for the case $n_i = 4$, $1 \le i \le m$, $m = 6$.*

Then

$$\varepsilon_1 + \varepsilon_2 + \varepsilon_3 \le t_{\tilde{n}}^{(\tilde{\mathcal{M}})}\varepsilon \le \varepsilon\,. \tag{3.6}$$

In Figure 1, the areas of $\tilde{A}$ which have contributed to $\varepsilon_1\Sigma^2$, $\varepsilon_2\Sigma^2$, and $\varepsilon_3\Sigma^2$ are shaded □, ▨, and ■, respectively.

**3.2.1. The case $n_m = 1$.** In this case, the proof of the induction step immediately follows from the third assertion of the following proposition. One just has to set $\Gamma = \Sigma$ $(= S(A^{(0)}))$ and $\mu = t_{\tilde{n}}^{(\tilde{\mathcal{M}})}$.

The first and third assertions of this proposition are modifications of a similar result from [12].

PROPOSITION 3.2. [illegible] $A = (a_{pq})$ [illegible] $n \ge 3$, [illegible] $B = E_{n-1}^T A E_{n-1}$ [illegible] $r$ $1 \le r \le n-1$ [illegible] $A$ [illegible] $(1,n), (2,n), \ldots, (r,n)$ [illegible] $A^{(0)} = A, A^{(1)}, \ldots, A^{(n-1)}$ [illegible]

(i) $\left[\sum_{i=1}^{r} |a_{in}^{(i-1)}|^2\right]^{1/2} \ge \left[\frac{2^{1-r}}{r}\sum_{i=1}^{r}|a_{in}|^2\right]^{1/2} - \left[\frac{r-1}{4}\sum_{i=1}^{r}\sum_{k=1}^{i-1}|a_{ik}|^2\right]^{1/2}, r \le n-1.$

(ii) $\left[\sum_{i=r+1}^{t} |a_{in}^{(r)}|^2\right]^{1/2} \ge \left[\frac{2^{-r}}{t-r}\sum_{i=r+1}^{t}|a_{in}|^2\right]^{1/2} - \left[\frac{r}{2}\sum_{i=r+1}^{t}\sum_{k=1}^{r}|a_{ik}|^2\right]^{1/2}, t > r.$

(iii) [illegible] $0 \le \mu < 1$ $0 \le \alpha \le 1$ [illegible] $\Gamma$ [illegible]

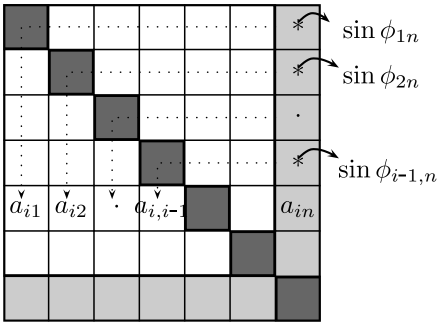
$$S^2(B) \le \mu(\alpha\Gamma^2) \quad [illegible] \quad \sum_{i=1}^{n-1}|a_{in}|^2 = (1-\alpha)\Gamma^2\,. \tag{3.7}$$

[illegible] $\mu'$ $0 \le \mu' < 1$ [illegible] $n$ [illegible] $\mu$ [illegible]

$$S^2(A^{(n-1)}) \le \mu'\,\Gamma^2\,.$$

[illegible] (i) Let us denote the rotation angle which is used to annihilate the element at position $(k,n)$ by $\phi_{kn}$. Consider the history of the element at position $(i,n)$, $i > 1$, up to its annihilation. We have

$$
\begin{aligned}
a_{in}^{(1)} &= \cos\phi_{1n}\, a_{in} + \sin\phi_{1n}\, a_{i1} \\
a_{in}^{(2)} &= \cos\phi_{2n}\, a_{in}^{(1)} + \sin\phi_{2n}\, a_{i2} \\
&\ \ \vdots \\
a_{in}^{(i-1)} &= \cos\phi_{i-1,n}\, a_{in}^{(i-2)} \\
&\qquad + \sin\phi_{i\text{-}1,n}\, a_{i,i-1}
\end{aligned}
$$



These relations imply

$$
\begin{aligned}
a_{in}^{(i-1)} &= a_{in}\cos\phi_{1n}\cos\phi_{2n}\cdots\cos\phi_{i-1,n} + a_{i,i-1}\sin\phi_{i-1,n} \\
&\quad + a_{i,i-2}\sin\phi_{i-2,n}\cos\phi_{i-1,n} + \cdots + a_{i1}\sin\phi_{1n}\cos\phi_{2n}\cdots\cos\phi_{i-1,n}\,.
\end{aligned}
$$

Note that $z = x + y$ implies $|z| \geq |x| - |y|$. Applying this to the above relation, one obtains

$$
\begin{aligned}
|a_{in}^{(i-1)}| &\geq \cos\phi_{1n}\cos\phi_{2n}\cdots\cos\phi_{i-1,n}|a_{in}| \\
&\qquad - \sum_{k=1}^{i-1} |a_{ik}|\,|\sin\phi_{kn}|\,\cos\phi_{k+1,n}\cdots\cos\phi_{i-1,n} \\
&\geq |a_{in}|\cdot(2^{-\frac{1}{2}})^{i-1} - \sum_{k=1}^{i-1} |a_{ik}|\,|\sin\phi_{kn}|\,.
\end{aligned}
$$

This inequality holds for $i = 1$ provided that an empty sum is defined as zero.

Summing up over $1 \leq i \leq r$ and using the Cauchy–Schwarz inequality, we obtain

$$
\begin{aligned}
\sum_{i=1}^{r} |a_{in}^{(i-1)}| &\geq \sum_{i=1}^{r} 2^{-\frac{i-1}{2}}|a_{in}| \;-\; \sum_{i=1}^{r}\sum_{k=1}^{i-1} |a_{ik}|\,|\sin\phi_{kn}| \\
&\geq 2^{-\frac{r-1}{2}} \sum_{i=1}^{r} |a_{in}| - \left[\sum_{i=1}^{r}\sum_{k=1}^{i-1} |a_{ik}|^2\right]^{1/2} \left[\frac{1}{2}\frac{r(r-1)}{2}\right]^{1/2} \\
&\geq \sqrt{2^{1-r}}\left[\sum_{i=1}^{r} |a_{in}|^2\right]^{1/2} - \frac{\sqrt{r(r-1)}}{2}\left[\sum_{i=1}^{r}\sum_{k=1}^{i-1} |a_{ik}|^2\right]^{1/2}.
\end{aligned}
$$

Since by the Cauchy–Schwarz inequality

$$
\sum_{i=1}^{r} |a_{in}^{(i-1)}| \leq \sqrt{r}\left[\sum_{i=1}^{r} |a_{in}^{(i-1)}|^2\right]^{1/2},
$$

the first assertion easily follows.

(ii) In this case we have for $r + 1 \leq i \leq t$, $t \leq n - 1$,

$$
\begin{aligned}
a_{in}^{(r)} &= a_{in}\cos\phi_{1n}\cos\phi_{2n}\cdots\cos\phi_{rn} + \ a_{ir}\sin\phi_{rn} + a_{i,r-1}\sin\phi_{r-1,n}\cos\phi_{rn} \\
&\quad + \cdots + a_{i1}\sin\phi_{1n}\cos\phi_{2n}\cdots\cos\phi_{rn}
\end{aligned}
$$

and

$$|a_{in}^{(r)}| \geq \cos\phi_{1n}\cdots\cos\phi_{rn}|a_{in}| - \sum_{k=1}^{r}|a_{ik}|\,|\sin\phi_{kn}|\,\cos\phi_{k+1,n}\cdots\cos\phi_{rn}$$

$$\geq |a_{in}|\cdot(2^{-\frac{1}{2}})^r - \sum_{k=1}^{r}|a_{ik}|\,|\sin\phi_{kn}|\,.$$

Hence

$$\sum_{i=r+1}^{t}|a_{in}^{(r)}| \geq \sum_{i=r+1}^{t}2^{-\frac{r}{2}}|a_{in}| \;-\; \sum_{i=r+1}^{t}\sum_{k=1}^{r}|a_{ik}|\,|\sin\phi_{kn}|$$

$$\geq 2^{-\frac{r}{2}}\left[\sum_{i=r+1}^{t}|a_{in}|^2\right]^{1/2} - \left[\sum_{i=r+1}^{t}\sum_{k=1}^{r}|a_{ik}|^2\right]^{1/2}\left[\frac{r(t-r)}{2}\right]^{1/2},$$

and an application of the Cauchy–Schwarz inequality yields (ii).

(iii) Using (i) with $r = n-1$ and the assumption (3.7), we obtain

$$\left[\sum_{i=1}^{n-1}|a_{in}^{(i-1)}|^2\right]^{1/2} \geq \left\{\left[\frac{2^{2-n}}{n-1}\right]^{1/2}\sqrt{1-\alpha} - \frac{\sqrt{n-2}}{2}\sqrt{\alpha}\right\}\Gamma\,. \tag{3.8}$$

The function

$$f_n(\alpha) = \left[\frac{2^{2-n}}{n-1}\right]^{1/2}\sqrt{1-\alpha} - \frac{\sqrt{n-2}}{2}\sqrt{\alpha}$$

is continuous and decreasing on the segment $[0,1]$ with the only root

$$\xi_n = \frac{1}{1+2^{n-4}(n-1)(n-2)}\,.$$

In the above relation, note that $n \geq 3$. If we take

$$\zeta_n = \frac{\xi_n}{2} = \frac{1}{2+2^{n-3}(n-1)(n-2)}\,,$$

we obtain from the relation (3.8)

$$\sum_{i=1}^{n-1}|a_{in}^{(i-1)}|^2 \geq \eta_n\Gamma^2\,, \quad \eta_n = [f_n(\zeta_n)]^2 > 0, \quad \alpha \in [0,\zeta_n]\,.$$

Thus, during the last $n-1$ steps, $S^2(A) = S^2(B) + (1-\alpha)\Gamma^2 \leq \mu(\alpha\Gamma^2) + (1-\alpha)\Gamma^2 = [1-\alpha(1-\mu)]\Gamma^2 \leq \Gamma^2$ has decreased by an amount not smaller than $\eta_n\Gamma^2$, $\eta_n > 0$. Hence

$$S^2(A^{(n-1)}) \leq \begin{cases} (1-\eta_n)\Gamma^2 & \text{if } \alpha \in [0,\zeta_n], \\ [1-\alpha(1-\mu)]\Gamma^2 & \text{if } \alpha \in [\zeta_n\,,\,1]. \end{cases}$$

So, we can complete the proof by setting

$$\mu' = \max\{1-\zeta_n(1-\mu)\,,\,1-\eta_n\}\,.$$

Note that $\mu'$ depends only on $n$ and $\mu$. □

3.3. Proposition 3.2(iii) provides a basis for the simplest proof of global convergence of the serial symmetric Jacobi method. Indeed, the proof uses induction with respect to $n$. The induction basis takes advantage of the fact that for $n = 2$, $S(A^{(1)}) = 0$. The induction hypothesis assumes that $S(A^{(\frac{(n-1)(n-2)}{2})}) \leq \mu S(A)$ for any initial symmetric matrix of order $n-1$, where $0 \leq \mu < 1$ depends only on $n-1$ (that is, on $n$). The induction step is then proved by Proposition 3.2(iii) and the assumption that $\Gamma = S(A)$.

However, the proof presented here can easily be generalized for strategies slightly more general than the column-cyclic strategy. We can consider any strategy which annihilates the elements in columnwise fashion so that in the $j$th column, $2 \leq j \leq n$, the annihilation ordering is $(\pi_{j-1}(1), j), (\pi_{j-1}(2), j), \ldots, (\pi_{j-1}(j-1), j)$ for any permutation $\pi_{j-1}$ of the set $\{1, 2, \ldots, j-1\}$. The proof of the induction step is almost the same as that of Proposition 3.2. Just consider the history of the element at position $(\pi_{n-1}(i), n)$, $i > 1$, up to its annihilation. This means that at step $i-1$ (of the last $n-1$ steps) the element at position $(\pi_{n-1}(i), n)$ is annihilated. Now, follow the proof of Proposition 3.2(iii) and replace all the subscripts $i$ and $k$ (but not the superscript $i$) with $\pi(i)$ and $\pi(k)$, respectively.

3.4. The first proof of global convergence of the serial methods is given by Forsythe and Henrici [13] and the first estimate of the form $S(A^{(N)}) \leq t_n S(A)$ by Henrici and Zimmermann [14] who obtained the known estimate

$$t_n = \left[ 1 - \prod_{i=1}^{n-2} \prod_{j=i+2}^{n} \cos\phi_{ij}^2 \right]^{1/2}.$$

See also similar results in [16, 10, 7].

### 3.2.2. The case $n_m \geq 2$.

Note that $m \geq 3$, $n \geq m$, and $n_m \geq 2$; hence $n \geq 4$. We assume $\Sigma > 0$, since otherwise there is nothing to prove. According to the pivot strategy, during the subsequent

$$\mu = n_m(\tilde{n} - n_{m-1}) + n_m(n_m - 1)/2 \tag{3.9}$$

Jacobi steps defined by the pivot pairs sequence $\mathsf{c}_{m1}, \mathsf{c}_{m2}, \mathsf{c}''_{m2}, \ldots, \mathsf{c}_{m,n_m}, \mathsf{c}''_{m,n_m}$, we have

$$\|A_{m-1,m}^{(\tilde{M}+k)}\|_F^2 + \sum_{i=1}^{m-2} \|A_{i,m-1}^{(\tilde{M}+k)}\|_F^2 = \|\tilde{A}_{m-1,m}\|_F^2 + \varepsilon_2 \Sigma^2, \quad 0 \leq k \leq \mu.$$

Therefore, for $0 \leq k \leq \mu$, we split $A^{(\tilde{M}+k)}$ into

$$A^{(\tilde{M}+k)} = H^{(\tilde{M}+k)} + T^{(\tilde{M}+k)},$$

where

$$H^{(\tilde{M}+k)} = \begin{bmatrix} B_{\tilde{n}-n_{m-1}}^{(\tilde{M}+k)} & 0 & \Gamma^{(\tilde{M}+k)} \\ 0 & 0 & 0 \\ [\Gamma^{(\tilde{M}+k)}]^T & 0 & A_{mm}^{(\tilde{M}+k)} \end{bmatrix}, \quad \Gamma^{(\tilde{M}+k)} = \begin{bmatrix} A_{1m}^{(\tilde{M}+k)} \\ \vdots \\ A_{m-2,m}^{(\tilde{M}+k)} \end{bmatrix}, \quad 0 \leq k \leq \mu.$$

Since the off-norm of $T^{(\tilde{M}+k)}$ is invariant under Jacobi transformations during all these $\mu$ steps, we shall use the matrix

$$W^{(\tilde{M}+k)} = \begin{bmatrix} B_{\tilde{n}-n_{m-1}}^{(\tilde{M}+k)} & \Gamma^{(\tilde{M}+k)} \\ [\Gamma^{(\tilde{M}+k)}]^T & A_{mm}^{(\tilde{M}+k)} \end{bmatrix}, \quad 0 \le k \le \mu . \tag{3.10}$$

Since the annihilations advance in a block-columnwise fashion, we have

$$(1-\varepsilon)\Sigma^2 = S^2(A_{mm}) + \sum_{i=1}^{m-1} \|A_{im}\|_F^2 = S^2(\tilde{A}_{mm}) + \sum_{i=1}^{m-1} \|\tilde{A}_{im}\|_F^2 . \tag{3.11}$$

This follows from the fact that the Frobenius norm is invariant under orthogonal transformations. Recall that $\varepsilon$ can be arbitrarily small. Because of (3.11), we consider the following two cases:

$$\text{(a) } S^2(\tilde{A}_{mm}) + \sum_{i=1}^{m-2} \|\tilde{A}_{im}\|_F^2 \ge \frac{2^{-n}}{1-2^{-2n}}\Sigma^2,$$

$$\text{(b) } \|\tilde{A}_{m-1,m}\|_F^2 > \left(1 - \varepsilon - \frac{2^{-n}}{1-2^{-2n}}\right)\Sigma^2.$$

We start with case (a).

(a) In this case, there exists $\jmath$, $1 \le \jmath \le n_m$, such that

$$\sum_{i\in\mathcal{N}_\jmath} |\tilde{a}_{i,\tilde{n}+\jmath}|^2 \ge 2^{-(2(n_m-\jmath)+1)n} \cdot \Sigma^2, \tag{3.12}$$

where

$$\mathcal{N}_j = \{1, 2, \dots, \tilde{n}+j-1\} \setminus \{\tilde{n}-n_{m-1}+1, \dots, \tilde{n}\}, \quad 1 \le j \le n_m ,$$

and $\tilde{A} = (\tilde{a}_{ij})$. Indeed, if (3.12) were false, then we would have

$$\begin{aligned} S^2(\tilde{A}_{mm}) + \sum_{i=1}^{m-2} \|\tilde{A}_{im}\|_F^2 &= \sum_{j=1}^{n_m} \sum_{i\in\mathcal{N}_j} |\tilde{a}_{i,\tilde{n}+j}|^2 < \sum_{j=1}^{n_m} 2^{-(2(n_m-j)+1)n}\Sigma^2 \\ &= \left(2^{-n} + 2^{-3n} + \cdots + 2^{-(2n_m-1)n}\right)\Sigma^2 = \frac{2^{-n}}{1-2^{-2n}}(1-2^{-2n_m n})\,\Sigma^2 \\ &< \frac{2^{-n}}{1-2^{-2n}}\,\Sigma^2 , \end{aligned}$$

which contradicts (a). If there are more than one $\jmath$ satisfying (3.12), we define $\jmath$ as the [illegible].

Let $\hat{A} = (\hat{a}_{ij})$, $\hat{A} = A^{(\hat{M})}$, $\hat{W} = W^{(\hat{M})}$, where $W^{(\hat{M})}$ is given by (3.10), and

$$\hat{M} = \tilde{M} + (\tilde{n}-n_{m-1}) + (\tilde{n}-n_{m-1}+1) + \cdots + (\tilde{n}-n_{m-1}+\jmath-2) .$$

In other words, $\hat{A}$ is the matrix iterate at the stage just before the annihilations in column $\jmath$ of the $m$th block-column have begun. Thus, $\hat{A}$ is obtained from $\tilde{A}$ by applying

$$\mu_\jmath = (\jmath-1)(\tilde{n}-n_{m-1}) + (\jmath-2)(\jmath-1)/2$$

additional Jacobi steps. They are defined by the sequence of pivot pairs $\mathsf{c}_{m1}, \mathsf{c}_{m2}, \mathsf{c}''_{m2}, \ldots, \mathsf{c}_{m,\jmath-1}, \mathsf{c}''_{m,\jmath-1}$. Now, consider the matrices $\hat{W}_{\tilde{n}-n_{m-1}+\jmath}$ and $W^{(k)}_{\tilde{n}-n_{m-1}+\jmath}$, where

$$\hat{W}_i = E_i^T \hat{W} E_i \quad \text{and} \quad W_i^{(k)} = E_i^T A^{(k)} E_i, \quad k \geq \tilde{M}\,.$$

Since all the Jacobi steps involved in the transition from $\tilde{A}$ to $\hat{A}$ cannot increase the off-norm of any $W^{(k)}_{\tilde{n}-n_{m-1}+\jmath-1}$, $\tilde{M} \leq k \leq \hat{M}$, we have

$$\begin{aligned} S^2(\hat{W}_{\tilde{n}-n_{m-1}+\jmath-1}) &\leq S^2(W^{(\tilde{M})}_{\tilde{n}-n_{m-1}+\jmath-1}) \leq S^2(\tilde{B}_{\tilde{n}-n_{m-1}}) + \sum_{j=1}^{\jmath-1} \sum_{i\in\mathcal{N}_j} |\tilde{a}_{i,\tilde{n}+j}|^2 \\ &\leq \varepsilon_1 \Sigma^2 + \left(2^{-(2n_m-1)n} + 2^{-(2n_m-3)n} + \cdots + 2^{-(2(n_m-\jmath)+3)n}\right)\Sigma^2 \\ &\leq \left(\varepsilon_1 + \frac{2^{(-2n_m+2\jmath-3)n}}{1-2^{-2n}}\right)\Sigma^2\,. \end{aligned} \tag{3.13}$$

We also know that none of these $\mu_\jmath$ Jacobi steps has changed the sum of squares of the affected elements in the last column of $W^{(\tilde{M}+k)}_{\tilde{n}-n_{m-1}+\jmath}$, $\tilde{M} \leq k \leq \hat{M}$. Therefore,

$$\sum_{i\in\mathcal{N}_\jmath} |\hat{a}_{i,\tilde{n}+\jmath}|^2 = \sum_{i\in\mathcal{N}_\jmath} |\tilde{a}_{i,\tilde{n}+\jmath}|^2\,. \tag{3.14}$$

In order to estimate the contribution to the off-norm reduction, coming from the annihilations in the $(\tilde{n}-n_{m-1}+\jmath)$th column, we apply Proposition 3.2(i) to $\hat{W}_{\tilde{n}-n_{m-1}+\jmath}$ with $r = \tilde{n} - n_{m-1} + \jmath - 1$. This, together with (3.12), (3.13), and (3.14), implies

$$\begin{aligned} \left[\sum_{i\in\mathcal{N}_\jmath} |\hat{a}^{(\hat{M}+i-1)}_{i,\tilde{n}+\jmath}|^2\right]^{1/2} &\geq \frac{2}{\sqrt{(\tilde{n}-n_{m-1}+\jmath-1)2^{\tilde{n}-n_{m-1}+\jmath}}} \left[\sum_{i\in\mathcal{N}_\jmath} |\hat{a}_{i,\tilde{n}+\jmath}|^2\right]^{1/2} \\ &\quad - \frac{\sqrt{\tilde{n}-n_{m-1}+\jmath-2}}{2}\left[\varepsilon_1 + \frac{2^{(-2n_m+2\jmath-3)n}}{1-2^{-2n}}\right]^{1/2} \Sigma \\ \geq \frac{2\cdot\; 2^{-(n_m-\jmath)n-\frac{n}{2}} \cdot 2^{-\frac{n-n_{m-1}}{2}}}{\sqrt{n-n_{m-1}-1}}\Sigma &- \frac{\sqrt{n-n_{m-1}-2}}{2}\left[\varepsilon_1 + \frac{2^{(-2n_m+2\jmath-3)n}}{1-2^{-2n}}\right]^{1/2} \Sigma \\ \geq \left\{\frac{2\sqrt{2}\cdot 2^{-(n_m-\jmath+1)n}}{\sqrt{n-n_{m-1}}} \right. &\left. - \frac{\sqrt{n-n_{m-1}}}{2}\left[\varepsilon_1 + \frac{2^{(-2n_m+2\jmath-3)n}}{1-2^{-2n}}\right]^{1/2}\right\}\Sigma\,. \end{aligned}$$

We can write

$$\left[\sum_{i\in\mathcal{N}_\jmath} |\hat{a}^{(\hat{M}+i-1)}_{i,\tilde{n}+\jmath}|^2\right]^{1/2} \geq g_{n,n_m,n_{m-1},\jmath}(\varepsilon_1)\Sigma,$$

where

$$g_{n,n_m,n_{m-1},\jmath}(x) = \frac{\sqrt{n-n_{m-1}}}{2}\left\{\frac{4\sqrt{2}\cdot 2^{-(n_m-\jmath+1)n}}{n-n_{m-1}} - \left[x + \frac{2^{(-2n_m+2\jmath-3)n}}{1-2^{-2n}}\right]^{1/2}\right\}.$$

The function $g_{n,n_m,n_{m-1},\jmath}$ is decreasing and convex for $0 \le x \le 1$,

$$g_{n,n_m,n_{m-1},\jmath}(0) = \frac{\sqrt{n-n_{m-1}}}{2} 2^{-(n_m-\jmath+1)n} \left[ \frac{4\sqrt{2}}{n-n_{m-1}} - \frac{2^{-\frac{n}{2}}}{\sqrt{1-2^{-2n}}} \right]$$
$$> \frac{\sqrt{3}}{2} 2^{-(n-2)n} \left[ \frac{4\sqrt{2}}{n-1} - 2^{-\frac{n}{2}} \right] > \frac{4}{n} 2^{-(n-2)n} ,$$

and with the positive root

$$\eta_{n,n_m,n_{m-1},\jmath} = 2^{(-2n_m+2\jmath-2)n} \left( \frac{32}{(n-n_{m-1})^2} - \frac{2^{-n}}{1-2^{-2n}} \right) .$$

Let

$$\zeta_{n,n_m,\jmath} = \frac{16}{n^2} \cdot 2^{-2(n_m-\jmath+1)n} .$$

Then

$$\frac{1}{2} \eta_{n,n_m,n_{m-1},\jmath} > \zeta_{n,n_m,\jmath} > \frac{16}{n^2}\, 2^{-2(n-2)n} \equiv \xi_n ,$$

and we have

$$\sum_{i\in\mathcal{N}_\jmath} |\hat{a}^{(\hat{M}+i-1)}_{i,\tilde{n}+\jmath}|^2 \ge [g_{n,n_m,n_{m-1}}(\zeta_{n,n_m,\jmath})]^2 \Sigma^2 \quad \text{whenever} \quad 0 \le \varepsilon_1 \le \zeta_{n,n_m,\jmath} .$$

Note that

$$[g_{n,n_m,n_{m-1}}(\zeta_{n,n_m,\jmath})]^2 = \frac{n-n_{m-1}}{4} 2^{-2(n_m-\jmath+1)n}$$
$$\times \left\{ \frac{2\sqrt{2}}{n-n_{m-1}} - \frac{4}{n} \left[ 1 - \frac{n^2}{16} \frac{2^{-n}}{1-2^{-2n}} \right]^{1/2} \right\}^2$$
$$> \frac{1}{2}\, 2^{-2(n-2)n} \left\{ \frac{2\sqrt{2}}{n-n_{m-1}} - \frac{4}{n} \left[ 1 - \frac{n^2}{16} \frac{2^{-n}}{1-2^{-2n}} \right]^{1/2} \right\}^2 .$$

Hence if $0 \le \varepsilon_1 \le \zeta_{n,n_m,\jmath}$, we have

$$\sum_{i\in\mathcal{N}_\jmath} |\hat{a}^{(\hat{M}+i-1)}_{i,\tilde{n}+\jmath}|^2 > \varrho_n \Sigma^2, \qquad \rho_n = \frac{8}{n^2} \left\{ \sqrt{2} - \left[ 1 - \frac{n^2}{16} \frac{2^{-n}}{1-2^{-2n}} \right]^{1/2} \right\}^2 2^{-2(n-2)n} .$$

This inequality certainly holds for $0 \le \varepsilon_1 \le \xi_n$. Thus, during the last $M - \tilde{M}$ Jacobi steps, $S^2(\tilde{A}) = S^2(\tilde{B}) + (1-\varepsilon)\Sigma^2$ has decreased by an amount not smaller than $\varrho_n \Sigma^2$, $\varrho_n > 0$, provided that $\varepsilon \le \xi_n$. Hence

$$S^2(A^{(M)}) \le \begin{cases} (1-\varrho_n)\Sigma^2 & \text{if } \ \varepsilon \in [0, \xi_n], \\ [1 - \varepsilon(1 - t_{\tilde{n}}^{(\tilde{\mathcal{M}})})]\Sigma^2 & \text{if } \ \varepsilon \in [\xi_n \, , \, 1]. \end{cases}$$

So, we can complete the proof of the theorem by setting

$$t_n^{(\mathcal{M})} = \max\{[1 - \xi_n(1 - t_{\tilde{n}}^{(\tilde{\mathcal{M}})}) \, , \, 1 - \varrho_n\} .$$

(b) In this case we begin the proof by assuming

$$\varepsilon < \frac{2^{-n}}{1-2^{-4n}}\,. \tag{3.15}$$

This implies

$$\|\tilde{A}_{m-1,m}\|_F^2 > \left(1-\frac{2^{-n}}{1-2^{-4n}}-\frac{2^{-n}}{1-2^{-2n}}\right)\Sigma^2 > \left(1-\frac{2^{-n+1}}{1-2^{-2n}}\right)\Sigma^2\,. \tag{3.16}$$

Let $\tilde{F}=[\tilde{A}_{m-1,1},\tilde{A}_{m-1,2},\ldots,\tilde{A}_{m-1,m}]$, and let $[\tilde{f}_1,\tilde{f}_2,\ldots,\tilde{f}_n]$ be the column-partition of $\tilde{F}$. We claim that there exists $\jmath$, $1\le \jmath\le n_m$, not necessarily the same as above, such that

$$\|\tilde{f}_{\tilde{n}+\jmath}\| > \begin{cases} 2^{-2(n_m-\jmath)n}\,\Sigma & \text{if } 1\le \jmath\le n_m-1,\\ (1-2^{-n+1})\,\Sigma & \text{if } \jmath=n_m\,. \end{cases} \tag{3.17}$$

In fact, if (3.17) were false, we would have

$$\begin{aligned} \|\tilde{A}_{m-1,m}\|_F^2 &= \sum_{j=1}^{n_m}\|\tilde{f}_{\tilde{n}+j}\|^2 \le \left[2^{-4(n_m-1)n}+2^{-4(n_m-2)n}+\cdots+2^{-4n}\right]\Sigma^2\\ &+\left(1-2^{-n+2}+2^{-2n+2}\right)\Sigma^2 = \frac{1-2^{-4n_mn}}{1-2^{-4n}}\Sigma^2-(2^{-n+2}+2^{-2n+2})\Sigma^2\\ &=\left[1-2^{-n+2}\left(1-2^{-n}-\frac{2^{-3n-2}-2^{-4n_mn+n-2}}{1-2^{-4n}}\right)\right]\Sigma^2 < \left(1-\frac{2^{-n+1}}{1-2^{-2n}}\right)\Sigma^2\,, \end{aligned}$$

which contradicts (3.16).

Recall that $\mathsf{c}_{m1},\mathsf{c}_{m2},\mathsf{c}''_{m2},\mathsf{c}_{m3},\mathsf{c}''_{m3},\ldots,\mathsf{c}_{mn_m},\mathsf{c}''_{mn_m} \sim \mathsf{c}_{m1},\mathsf{c}_{m2},\mathsf{c}_{m3},\ldots,\mathsf{c}_{mn_m},$ $\mathsf{c}''_{m2},\mathsf{c}''_{m3},\ldots,\mathsf{c}''_{mn_m}$. Hence the same matrix $A^{(\tilde{M}+\mu)}$ (thus, the same block $A^{(\tilde{M}+\mu)}_{m-1,m}$, where $\mu$ is given by (3.9)) is obtained if we assume that the annihilation ordering is defined by $\mathsf{c}_{m1},\mathsf{c}_{m2},\mathsf{c}_{m3},\ldots,\mathsf{c}_{mn_m},\mathsf{c}''_{m2},\mathsf{c}''_{m3},\ldots,\mathsf{c}''_{mn_m}$. Accepting that, note that the rotations which annihilate the elements in the diagonal block $A^{(k)}_{mm}$ do not change the Frobenius norm of $A^{(k)}_{m-1,m}$.

Therefore, we divide the proof into two stages. In Stage I we estimate the weight remaining in $A^{(k)}_{m-1,m}$ after all Jacobi steps defined by the sequence $\mathsf{c}_{m1},\mathsf{c}_{m2},\ldots,\mathsf{c}_{mn_m}$ are completed. Actually, we shall compute a lower bound of $\|A^{(\tilde{M}+n_m(\tilde{n}-n_{m-1}))}_{m-1,m}\|_F = \|A^{(\tilde{M}+\mu)}_{m-1,m}\|_F$.

In Stage II, we use a similar proof as in case (a), but we apply it only to the part of the $m$th block-column consisting of the blocks $A^{(k)}_{m-1,m}$ and $A^{(k)}_{mm}$. In this stage, the annihilation ordering is defined by the sequence of pairs $\mathsf{c}'_{\jmath 1},\mathsf{c}'_{\jmath 2},\mathsf{c}''_{\jmath 2}\ldots,\mathsf{c}'_{\jmath n_\jmath},\mathsf{c}''_{\jmath n_\jmath}$.

Stage I. Let $\jmath$ be the first (smallest) index for which (3.17) holds. Let $\hat{A}=(\hat{a}_{ij})$ be the matrix $A^{(\hat{M})}$, where

$$\hat{M}=\tilde{M}+\tilde{\mu}_\jmath,\quad \tilde{\mu}_\jmath=(\tilde{n}-n_{m-1})(\jmath-1)\,.$$

Thus, $\hat{A}$ is obtained from $\tilde{A}$ by applying additional $\tilde{\mu}_\jmath$ Jacobi steps according to the sequence of pivot pairs $\mathsf{c}_{m1},\mathsf{c}_{m2},\ldots,\mathsf{c}_{m,\jmath-1}$. Let $\hat{F}=[\hat{A}_{m-1,1},\hat{A}_{m-1,2},\ldots,\hat{A}_{m-1,m}]$

and let $\hat F = [\hat f_1, \hat f_2, \dots, \hat f_n]$ be its column-partition. More generally, for $0 \le k \le \tilde\mu_\jmath$, let

$$\tilde F^{(k)} = [A_{m-1,1}^{(\tilde M+k)}, A_{m-1,2}^{(\tilde M+k)}, \dots, A_{m-1,m}^{(\tilde M+k)}] = [\tilde f_1^{(k)}, \tilde f_2^{(k)}, \dots, \tilde f_n^{(k)}].$$

The Jacobi transformations involved in the transition from $\tilde F$ to $\hat F$ do not change the Frobenius norm of any $\tilde F^{(k)} E_{\tilde n+\jmath-1}$, $0 \le k \le \tilde\mu_\jmath$, and they do not transform any element of $\tilde A_{m-1,m-1}$. Therefore, we have

$$\begin{aligned} &\sum_{j=1}^{\tilde n - n_{m-1}} \|\hat f_j\|^2 + \sum_{j=\tilde n+1}^{\tilde n+\jmath-1} \|\hat f_j\|^2 = \sum_{j=1}^{\tilde n - n_{m-1}} \|\tilde f_j\|^2 + \sum_{j=\tilde n+1}^{\tilde n+\jmath-1} \|\tilde f_j\|^2 \\ (3.18)\quad &+ \left(2^{-4(n_m-1)n} + \cdots + 2^{-(4(n_m-\jmath+1)n}\right)\Sigma^2 \le \left(\varepsilon_2 + \frac{2^{-4(n_m-\jmath+1)n}}{1-2^{-4n}}\right)\Sigma^2 .\end{aligned}$$

We know that none of these $\tilde\mu_\jmath$ Jacobi transformations has affected any of the elements of $\tilde f_{\tilde n+\jmath}^{(k)}$. Therefore,

$$\hat f_{\tilde n+\jmath} = \tilde f_{\tilde n+\jmath} . \tag{3.19}$$

We also know that the later transformations, defined by $\mathsf{c}_{m,\jmath+1}, \dots, \mathsf{c}_{m,n_m}$, will not affect the $\jmath$th column of $A_{m-1,m}^{(\hat M+\tilde n-n_{m-1})}$.

So, let us consider the transformations which do affect $\hat f_{\tilde n+\jmath}$, i.e., those which annihilate the elements at positions given in $\mathsf{c}_{m,\jmath}$.

For $1 \le k \le \tilde n - n_{m-1}$, let

$$\hat F^{(k)} = [A_{m-1,1}^{(\hat M+k)}, A_{m-1,2}^{(\hat M+k)}, \dots, A_{m-1,m}^{(\hat M+k)}] = [\hat f_1^{(k)}, \hat f_2^{(k)}, \dots, \hat f_n^{(k)}].$$

We can apply Proposition 3.2(ii) to $E_{\tilde n+\jmath}^T \hat A E_{\tilde n+\jmath}$, with $r = \tilde n - n_{m-1}$, to obtain

$$\begin{aligned} \|\hat f_{\tilde n+\jmath}^{(\tilde n-n_{m-1})}\| &\ge \left[\frac{2^{-(\tilde n-n_{m-1})}}{n_{m-1}}\right]^{1/2} \|\hat f_{\tilde n+\jmath}\| - \left[\frac{\tilde n-n_{m-1}}{2}\right]^{1/2} \left[\sum_{i=1}^{\tilde n-n_{m-1}} \|\hat f_i\|^2\right]^{1/2} \\ &\ge \frac{2^{-\frac{n-3}{2}}}{\sqrt{n_{m-1}}} \|\tilde f_{\tilde n+\jmath}\| - \left[\frac{\tilde n-n_{m-1}}{2}\right]^{1/2} \left[\varepsilon_2 + \frac{2^{-4(n_m+1-\jmath)n}}{1-2^{-4n}}\right]^{1/2} \Sigma .\end{aligned}$$

Here we have used (3.19), (3.18), and $\tilde n - n_{m-1} = n - n_m - n_{m-1} \le n-2-1$. If we require

$$\varepsilon_2 + \frac{2^{-4(n_m+1-\jmath)n}}{1-2^{-4n}} \le 2 \cdot 2^{-4(n_m+1-\jmath)n}, \tag{3.20}$$

we obtain

$$\|\hat f_{\tilde n+\jmath}^{(\tilde n-n_{m-1})}\| \ge \frac{2^{-\frac{n-3}{2}}}{\sqrt{n_{m-1}}} \|\tilde f_{\tilde n+\jmath}\| - \sqrt{\tilde n - n_{m-1}}\, 2^{-2(n_m+1-\jmath)n}\, \Sigma .$$

Now, we consider the two possible cases $\jmath < n_m$ and $\jmath = n_m$.

If $\jmath < n_m$, then (3.17) implies

$$\|\hat f_{\tilde n+\jmath}^{(\tilde n-n_{m-1})}\| \ge \frac{2^{-2(n_m+1-\jmath)n}}{\sqrt{n_{m-1}}} \left[2^{-\frac{n-3}{2}} 2^{2n} - \frac{\tilde n}{2}\right]\Sigma \ge \frac{2^{-2(n_m+1-\jmath)n}}{\sqrt{n_{m-1}}} \left[2^{\frac{3}{2}n+\frac{3}{2}} - \frac{n-2}{2}\right]\Sigma .$$

Here we have used the fact that $n_{m-1}(\tilde{n}-n_{m-1}) \leq \tilde{n}^2/4$ and $\tilde{n} \leq n-2$.

If $\jmath = n_m$, then (3.17) implies

$$\begin{aligned} \|\hat{f}_{\tilde{n}+\jmath}^{(\tilde{n}-n_{m-1})}\| &\geq \frac{2^{-\frac{n-3}{2}}(1-2^{-n+1})}{\sqrt{n_{m-1}}}\Sigma - \sqrt{\tilde{n}-n_{m-1}}\,2^{-2n}\Sigma \\ &\geq \frac{2^{-2n}}{\sqrt{n_{m-1}}}\left[\frac{7}{8}2^{\frac{3}{2}n+\frac{3}{2}} - \frac{n-2}{2}\right]\Sigma\,. \end{aligned}$$

Here we have used the inequality $2^{-n+1} \leq 1/8$ since $n \geq 4$.

Taking into account (3.20), we can conclude that, if

$$\varepsilon_2 \leq \frac{1-2^{-4n+1}}{1-2^{-4n}}2^{-4n_m n}, \tag{3.21}$$

then

$$\|\hat{f}_{\tilde{n}+\jmath}^{(\tilde{n}-n_{m-1})}\| \geq \alpha_{n,n_m,n_{m-1}}\Sigma\,, \quad \alpha_{n,n_m,n_{m-1}} = \frac{2^{-2n_m n}}{\sqrt{n_{m-1}}}\left[\frac{7\sqrt{2}}{4}2^{\frac{3}{2}n} - \frac{n-2}{2}\right]$$

holds for all possible choices of $\jmath$.

As we have noted, the later transformations, defined by $\mathsf{c}_{m,\jmath+1},\ldots,\mathsf{c}_{m,n_m}$, do not affect the $\jmath$th column of $A_{m-1,m}^{(\hat{M}+\tilde{n}-n_{m-1})}$, so $\tilde{f}_{\tilde{n}+\jmath}^{((\tilde{n}-n_{m-1})n_m)} = \hat{f}_{\tilde{n}+\jmath}^{(\tilde{n}-n_{m-1})}$. The subsequent $n_m(n_m-1)/2$ rotations which annihilate the off-diagonal elements of $A_{mm}^{(k)}$ do not change the Frobenius norm of $\tilde{A}^{((\tilde{n}-n_{m-1})n_m)}$. Therefore, if (3.21) holds, then

$$\|A_{m-1,m}^{(\bar{M})}\|_F \geq \alpha_{n,n_m,n_{m-1}}\Sigma\,, \tag{3.22}$$

where

$$\bar{M} = M - n_{m-1}n_m - \frac{n_m(n_m-1)}{2}\,.$$

We complete the considerations in Stage I by making some preparations for Stage II.

Let $r$ be the smallest positive integer such that

$$\alpha_{n,n_m,n_{m-1}}^2 \geq \frac{2^{-rn}}{1-2^{-2n}}\,.$$

Obviously, $r$ depends on $n_{m-1}$ and $n_m$. Since $n_m \leq n-2$, we have

$$\alpha_{n,n_m,n_{m-1}} \geq \frac{2^{-2(n-2)n}}{\sqrt{n-2}}\left[\frac{7\sqrt{2}}{4}\cdot 2^{\frac{3}{2}n} - \frac{n-2}{2}\right].$$

Running the following simple program (m-file) in MATLAB, one finds out that all `y-z` are positive. Since for larger $n$, $(n-2)/2$ is negligible compared to $2^{(3/2)n}$, we have shown that $r \leq 4n-10$.

```
format long e;
  for n=4:100
  x=2^(n);
  y=((7/4)*sqrt(2/(n-2))*x*sqrt(x)-sqrt(n-2)/2);
  z=x^(1.46)/sqrt(1-x^(-2));
  display([n,y,z,y-z])
end
```

Now, we can conclude that there is some $\jmath$, not necessarily the same as earlier, such that

$$\|\bar{f}_{\tilde{n}+\jmath}\|^2 \geq 2^{-(2(n_m-\jmath)+r)n}\,\Sigma^2\,. \tag{3.23}$$

Indeed, on the contrary we would have

$$\begin{aligned}\|A^{(\bar{M})}_{m-1,m}\|_F^2 &= \sum_{j=1}^{n_m}\|\bar{f}_{\tilde{n}+j}\|^2 < \sum_{j=1}^{n_m} 2^{-(2(n_m-\jmath)+r)n}\cdot\Sigma^2\\ &= 2^{-rn}\left[2^{-2(n_m-1)n}+2^{-2(n_m-2)n}+\cdots+1\right]\Sigma^2\\ &= 2^{-rn}\frac{1-2^{-2n_m n}}{1-2^{-2n}}\Sigma^2 < \frac{2^{-rn}}{1-2^{-2n}}\Sigma^2 \leq \alpha^2_{n,n_m,n_{m-1}}\Sigma^2\,,\end{aligned}$$

which contradicts (3.22). Let $\jmath$ be the smallest index satisfying (3.23).

For $1 \leq k \leq M-\bar{M}$, let $\bar{A}^{(k)} = (A_{ij}^{(\bar{M}+k)})$. Then $\bar{A} = \bar{A}^{(0)}$ is the matrix appearing at the beginning of Stage II.

[illegible] II Note that the rotations applied to $A^{(k)}$ for $\tilde{M} \leq k \leq \bar{M}$ have not changed any element of $A^{(k)}_{m-1,m-1}$, so $S^2(\bar{A}_{m-1,m-1}) = \varepsilon_3\Sigma^2$. Let $G^{(k)} = (g^{(k)}_{pq})$ with $1 \leq p, q \leq n_{m-1}+n_m$, be defined by

$$G^{(k)} = \begin{bmatrix} G^{(k)}_{11} & G^{(k)}_{12}\\ G^{(k)}_{21} & G^{(k)}_{22}\end{bmatrix} = \begin{bmatrix} \bar{A}^{(k)}_{m-1,m-1} & \bar{A}^{(k)}_{m-1,m}\\ \bar{A}^{(k)}_{m,m-1} & \bar{A}^{(k)}_{mm}\end{bmatrix}, \quad 0 \leq k \leq M-\bar{M},$$

and let $G = G^{(0)}$. We shall now estimate the contribution to the off-norm reduction coming from the last $M-\bar{M}$ Jacobi steps. We can restrict our attention to the matrices $G^{(k)}$, since the sum of squares of other off-diagonal elements of $\bar{A}^{(k)}$ remains invariant during these transformations.

The method applies Jacobi rotations in a columnwise fashion, following the ordering defined by $\mathsf{c}'_{j1}, \mathsf{c}'_{j2}, \mathsf{c}''_{j2}, \ldots, \mathsf{c}'_{jn_j}, \mathsf{c}''_{jn_j}$. Let

$$\nu_j = \nu_{j-1} + n_{m-1} + j - 1, \quad 1 \leq j \leq n_m, \quad \nu_0 = 0,$$

and let $G^{(k)}_j$ be the leading $j \times j$ submatrix of $G^{(k)}$. Let $g^{(k)}_2, g^{(k)}_3, \ldots, g^{(k)}_{n_{m-1}+n_m}$ be the columns of the strict uppertriangle of $G^{(k)}$, i.e. $g^{(k)}_j = [g^{(k)}_{1j}, g^{(k)}_{2j}, \ldots, g^{(k)}_{j-1,j}]^T$ and let $g_j = g^{(0)}_j$. Because of (3.23), it is obvious that

$$\|g_{n_{m-1}+\jmath}\|^2 \geq \|\bar{f}_{\tilde{n}+\jmath}\|^2 \geq 2^{-(2(n_m-\jmath)+r)n}\Sigma^2$$

holds. Note that $\jmath$ is the smallest integer satisfying the second inequality. Let us redefine $\jmath$ to be the smallest of the numbers in the set $\{1, 2, \ldots, n_m\}$, satisfying

$$\|g_{n_{m-1}+\jmath}\|^2 \geq 2^{-(2(n_m-\jmath)+r)n}\Sigma^2\,.$$

Since the Jacobi rotations cannot increase the off-norm, this assumption yields

$$S^2(G^{(\nu_{\jmath-1})}_{n_{m-1}+\jmath-1}) \leq S^2(G^{(0)}_{11}) + \sum_{j=1}^{\jmath-1}\|g_{n_{m-1}+j}\|^2 \leq \varepsilon_3\Sigma^2 + \frac{2^{-(2(n_m-\jmath+1)+r)n}}{1-2^{-2n}}\Sigma^2\,.$$

Here, the empty sum is considered zero.

Now, consider the contribution to the off-norm reduction coming from the annihilations in the vector $g_{n_{m-1}+\jmath}^{(\nu_\jmath-1)}$. Applying Proposition 3.2(i) to $G_{n_{m-1}+\jmath}^{(\nu_\jmath-1)}$, we obtain

$$\left[\sum_{i=1}^{n_{m-1}+\jmath-1} |g_{i,n_{m-1}+\jmath}^{(\nu_{\jmath-1}+i-1)}|^2\right]^{1/2} \geq \left\{\left[\frac{2^{-(n_{m-1}+\jmath-2)}}{n_{m-1}+\jmath-1} 2^{-(2(n_m-\jmath)+r)n}\right]^{1/2}\right.$$
$$\left. - \left[\frac{n_{m-1}+\jmath-2}{4}\left(\varepsilon_3 + \frac{2^{-(2(n_m-\jmath+1)+r)n}}{1-2^{-2n}}\right)\right]^{1/2}\right\}\Sigma \geq \frac{2^{-(n_m-\jmath+1+\frac{r}{2})n}}{\sqrt{n_{m-1}+n_m-1}}$$
$$\times\left\{2^{n+1-\frac{n_{m-1}+n_m}{2}} - \frac{n_{m-1}+n_m\text{-}1}{2}\left[2^{(2(n_m-\jmath+1)+r)n}\varepsilon_3 + \frac{1}{1-2^{-2n}}\right]^{1/2}\right\}\Sigma\,.$$

To simplify the expression under the square root, we can require

$$\varepsilon_3 \leq \frac{1-2^{-2n+1}}{1-2^{-2n}} 2^{-(2n_m+r)n}\,. \tag{3.24}$$

Then the expression under the square root is not larger than 2 and we have

$$\left[\sum_{i=1}^{n_{m-1}+\jmath-1} |g_{i,n_{m-1}+\jmath}^{(\nu_\jmath+i-1)}|^2\right]^{1/2} \geq \frac{2^{-(n_m-\jmath+1+\frac{r}{2})n}}{\sqrt{n_{m-1}+n_m}}\left[2^{\frac{n}{2}+1} - \frac{n_{m-1}+n_m}{\sqrt{2}}\right]\Sigma$$
$$\geq \frac{2^{-(n_m+\frac{r}{2})n}}{\sqrt{n_{m-1}+n_m}}\left[2^{\frac{n}{2}+1} - \frac{n_{m-1}+n_m}{\sqrt{2}}\right]\Sigma \equiv \sqrt{\theta_{n_{m-1},n_m}(n)}\,\Sigma > 0\,, \tag{3.25}$$

with $\theta_{n_{m-1},n_m}(n) > 0$. Looking back at the relations (3.5), (3.6), (3.15), (3.21), and (3.24), we can conclude that (3.25) holds, provided that

$$\varepsilon \leq 2^{-(2n_m+r+1)n}\,.$$

Thus, during the last $M - \bar{M}$ Jacobi steps, $S^2(\bar{A})$ has decreased by the amount not smaller than $\theta_{n_{m-1},n_m}(n)\Sigma^2$. Hence

$$S^2(A^{(M)}) \leq \begin{cases} [1-\theta_{n_{m-1},n_m}(n)]\,\Sigma^2 & \text{if } \varepsilon \in [0, 2^{(-2n_m+r+1)n}], \\ [1-\varepsilon(1-t_{\tilde{n}}^{(\tilde{\mathcal{M}})})]\,\Sigma^2 & \text{if } \varepsilon \in [2^{(-2n_m+r+1)n}\,,\, 1]. \end{cases}$$

Since $r < 4n - 10$, we can complete the proof by setting

$$t_n^{(\mathcal{M})} = \max\{[1 - 2^{-(4n+2n_m-9)n}(1 - t_{\tilde{n}}^{(\tilde{\mathcal{M}})})\,,\, 1 - \theta_{n_{m-1},n_m}(n)\}\,.$$

[illegible] 3.5. Theorem 2.1 holds even if in [6, Algorithm 1] one dynamically combines the cases $k = 0$ and $k = 1$, that is to say if some diagonal blocks are operated twice and some are operated once. Indeed, the proof can use the same induction per block-columns. If $A_{rr}$ is operated once, then the appropriate equivalent pivot strategy within this block-column is simply the columnwise one and the proof of the induction step is similar to case (a).

**Acknowledgments.** The author is thankful to Z. Drmač and D. Kressner, and especially to N. J. Higham and the anonymous referees for reading and improving the paper.

## REFERENCES

[1] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[2] J. J. Dongarra, I. Duff, D. C. Sorensen, and H. A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia, 1991.

[3] Z. Drmač, *Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic*, SIAM J. Sci. Comput., 18 (1997), pp. 1200–1222.

[4] Z. Drmač, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.

[5] Z. Drmač and K. Veselić, *New Fast and Accurate Jacobi SVD Algorithm:* I, LAPACK Working Note 169, 2005. Available online at http://www.netlib.org/lapack/lawns/downloads/

[6] Z. Drmač and K. Veselić, *New Fast and Accurate Jacobi SVD Algorithm:* II, LAPACK Working Note 170, 2005. Available online at http://www.netlib.org/lapack/lawns/downloads/

[7] K. V. Fernando, *Linear convergence of the row cyclic Jacobi and Kogbetliantz methods*, Numer. Math., 56 (1989), pp. 73–94.

[8] E. R. Hansen, *On Jacobi Methods and Block Jacobi Methods for Computing Matrix Eigenvalues*, Ph.D. thesis, Stanford University, Stanford, CA, 1960.

[9] E. R. Hansen, *On cyclic Jacobi methods*, SIAM J. Appl. Math., 11 (1963), pp. 448–459.

[10] V. Hari, *On the convergence of cyclic Jacobi-like processes*, Linear Algebra Appl., 81 (1986), pp. 105–127.

[11] V. Hari, *On sharp quadratic convergence bounds for the serial Jacobi methods*, Numer. Math., 60 (1991), pp. 375–406.

[12] V. Hari and K. Veselić, *On Jacobi methods for singular value decompositions*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 741–754.

[13] G. E. Forsythe and P. Henrici, *The cyclic Jacobi method for computing the principal values of a complex matrix,* Trans. Amer. Math. Soc., 94 (1960), pp. 1–23.

[14] P. Henrici and K. Zimmermann, *An estimate for the norms of certain cyclic Jacobi operators*, Linear Algebra Appl., 1 (1968), pp. 489–501.

[15] W. Mascarenhas, *On the convergence of the Jacobi method for arbitrary orderings*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1197–1209.

[16] L. Nazareth, *On the convergence of the cyclic Jacobi method*, Linear Algebra Appl., 12 (1975), pp. 151–164.

[17] N. Rhee and V. Hari, *On the global and cubic convergence of a quasi-cyclic Jacobi method*, Numer. Math., 66 (1993), pp. 97–122.

[18] G. Shroff and R. Schreiber, *On the convergence of the cyclic Jacobi method for parallel block orderings*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 326–346.

[19] K. Veselić and V. Hari, *A note on a one-sided Jacobi algorithm*, Numer. Math., 56 (1989), pp. 627–633.

[20] J. Wilkinson, *Note on the quadratic convergence of the cyclic Jacobi process*, Numer. Math., 4 (1962), pp. 296–300.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 370–376

# PATH PRODUCT MATRICES AND EVENTUALLY INVERSE $M$-MATRICES*

CHARLES R. JOHNSON† AND RONALD L. SMITH‡

**Abstract.** Those nonnegative matrices, some Hadamard power of which are inverse $M$-matrices, are characterized. This requires a refinement of the strict path product necessary condition [C. R. Johnson and R. L. Smith, *Linear Multilinear Algebra*, 46 (1999), pp. 177–191] for an inverse $M$-matrix. The smallest such Hadamard power may be arbitrarily large. It is also shown that, beyond some threshold, *all* continuous Hadamard powers of an inverse $M$-matrix are inverse $M$. In the process, several new results about inverse $M$-matrices are given.

**Key words.** $M$-matrices, inverse $M$-matrices, path product matrices

**AMS subject classifications.** 15A48, 15A45

**DOI.** 10.1137/050636048

## 1. Path product matrices and eventually inverse $M$-matrices.

An *inverse* $M$ ($IM$) matrix is an invertible $n$-by-$n$ nonnegative matrix whose inverse has nonpositive off-diagonal entries [2, 8, 9]. The $M$-matrices (those square matrices with positive principal minors and nonpositive off-diagonal entries) comprise the inverse class of $IM$ matrices and have a wide variety of applications. For example, $M$-matrices arise in various aspects of numerical linear algebra, in cost model matrices in economics, and in the numerical solution of certain types of differential equations. In addition to their obvious application to inverse problems involving $M$-matrices, $IM$ matrices themselves arise in a number of applications such as numerical integration [15], the Ising model of ferromagnetism [15], taxonomy [1], and random energy models in statistical physics [6]. Many applications of $IM$ matrices involve the subclass of strictly ultrametric matrices. Consequently, there has been a great deal of work on this particular type of $IM$ matrix (see, for example, [4, 5, 7, 13]). However, other special types of $IM$ matrices have also been considered (see [12, 14]).

An $n$-by-$n$ entry wise nonnegative matrix $A = (a_{ij})$ is called a *strict path product* ($SPP$) *matrix* if, for any triple of indices $i, j, k \in N = \{1, 2, \ldots, n\}$,

$$\frac{a_{ij}a_{jk}}{a_{jj}} \leq a_{ik} \tag{1}$$

with strict inequality whenever $i \neq j$ and $k = i$ [10]. We say that a nonnegative matrix is *normalized* if it has ones on the diagonal and off-diagonal entries less than 1. In the event that $A$ is a normalized $SPP$ matrix, the basic path product inequalities become

$$a_{ij}a_{jk} \leq a_{ik} \tag{2}$$

(with strict inequality whenever $i \neq j$ and $k = i$).

---

*Received by the editors July 14, 2005; accepted for publication (in revised form) by H. J. Woerdeman October 11, 2006; published electronically March 19, 2007.
http://www.siam.org/journals/simax/29-2/63604.html

†Department of Mathematics, College of William and Mary, Williamsburg, VA 23187 (crjohnso@math.wm.edu). The work of this author was supported, in part, by National Science Foundation grant DMS 92–00899 and by Office of Naval Research contract N00014–90–J1739.

‡Department of Mathematics, University of Tennessee at Chattanooga, 615 McCallie Avenue, Chattanooga, TN 37403-2598 (ronald-smith@utc.edu). The work of this author was supported, in part, by the Center of Excellence in Computer Applications at the University of Tennessee at Chattanooga and by the University of Chattanooga Foundation and was completed while the author was on sabbatical in 2004–2005.

In [10], it was noted that any $IM$ matrix is $SPP$ and that for $n \leq 3$ (but not greater) the two classes are the same. See also [15]. Our interest here lay in further relating these two classes via consideration of Hadamard powers: by the $p$th Hadamard power $A^{(p)}$ of $A$ we mean the matrix $(a_{ij}^p)$ for a real number $p$. We were motivated, in part, by the main result of a recent paper [3] in which it was shown that if $A$ is $IM$, then $A^{(k)}$ is $IM$ for any positive integer $k \geq 1$. If $A$ is $SPP$ and $p > 0$, it is clear that $A^{(p)}$ remains $SPP$. (In fact, a Hadamard product of any two $SPP$ matrices is again $SPP$, and this remains so for the variants of $SPP$ to be mentioned in the following.) This raises the question as to whether $A^{(p)}$ eventually becomes $IM$ as $p$ increases without bound. If there is a $P > 0$ such that $A^{(p)}$ is $IM$ for all $p > P$, we call $A$ an . Our question, then, is: which nonnegative matrices are $EIM$? It is clear that it is necessary that an $EIM$ matrix be $SPP$, but it is not sufficient.

1. Consider the normalized $SPP$ matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0.5 & 0.7 & 0.4 \\ 0.5 & 1 & 0.5 & 0.25 \\ 0.7 & 0.5 & 1 & 0.5 \\ 0.4 & 0.25 & 0.5 & 1 \end{bmatrix}.$$

Since the $2,4$ cofactor of $A^{(p)}$ is $c_{24}^{(p)} = [(0.5)^p - (0.35)^p][(0.4)^p - (0.35)^p]$, which is positive for all $p > 0$, we see that $A$ is not $EIM$.

We also address the question of whether there is some converse to the statement that $IM$ implies (some kind of) $SPP$.

In order to answer these questions, we refine further the path product conditions for $IM$ matrices and identify the appropriate additional necessary conditions. These involve the arrangement of occurrences of equality in the path product inequalities (1). We say that $(i, j, k)$ is a (for the $SPP$ matrix $A$) if equality occurs in (1). For instance, $(2, 3, 4)$ is a path product equality triple in Example 1. Note that the path product equality triples of $A$ and a normalized version of $A$ are the same. When $n \geq 4$, we will see that path product equalities in an $IM$ matrix (which can occur) necessarily imply others. In view of previous work [11], this is not surprising, as a path product equality implies that a 2-by-2 almost principal minor is 0, so that a certain 3-by-3 principal submatrix has a 0 in its inverse. By [11] this implies further 0's in the inverse of the full matrix (though it is not necessary that all inverse 0's stem from path product equalities).

2. Consider the $IM$ matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0.4 & 0.4 & 0.3 \\ 0.4 & 1 & 0.5 & 0.5 \\ 0.4 & 0.4 & 1 & 0.6 \\ 0.4 & 0.4 & 0.4 & 1 \end{bmatrix}.$$

Matrix $A$ contains no path product equalities. However, $A^{-1}$ has a 0 in the $1,4$ position.

It was noted in [10, Corollary 2.6] that if $A$ is an $n$-by-$n$ $SPP$ matrix, then there exist positive diagonal matrices $D$ and $E$ such that $B = DAE$, in which $B$ is a normalized $SPP$ matrix. Thus, $B^{(p)} = D^p A^{(p)} E^p$ for $p \geq 0$ and, since it is apparent that $SPP$ matrices are closed under positive diagonal equivalence, $A^{(p)}$ is $SPP$ if and only if $B^{(p)}$ is. Therefore, it suffices to consider normalized $SPP$ matrices when studying (positive) Hadamard powers of $SPP$ matrices. In that $IM$ matrices are closed under positive diagonal equivalence, the same can be said for $IM$ matrices.

First, we identify the case in which no path product equalities occur, i.e., all inequalities (1) are strict, and call such $SPP$ matrices *totally strict path product* (*totally* $SPP$). Observe that totally $SPP$ matrices are necessarily positive. Totally $SPP$ is not necessary for $EIM$ (just consider the $IM$ matrix $B = A[\{2,3,4\}]$ of Example 1), but we will see (Theorem 3) that totally $SPP$ is sufficient for $EIM$.

Consider the following condition (3) on the collection of path product inequalities: for all distinct indices $i, j, k \in N$ and for all $m \in N - \{i, j, k\}$,

$$a_{ik} = \frac{a_{ij}a_{jk}}{a_{jj}} \quad \text{implies that either} \quad a_{im} = \frac{a_{ij}a_{jm}}{a_{jj}} \quad \text{or} \quad a_{mk} = \frac{a_{mj}a_{jk}}{a_{jj}}. \tag{3}$$

If (3) is satisfied by an $SPP$ matrix, we say that $A$ is *purely strict path product* (*purely* $SPP$). We will see (Lemma 1) that, in purely $SPP$ matrices, path product equalities force certain cofactors to vanish. Notice that condition (3) does not hold for Example 1 since $a_{24} = 0.25 = (0.5)(0.5) = a_{23}a_{34}$ while $a_{14} = 0.4 > (0.7)(0.5) = a_{13}a_{34}$ and $a_{21} = 0.5 > (0.5)(0.7) = a_{23}a_{31}$. We show that any $IM$ matrix is purely $SPP$ (Theorem 1) and that $EIM$ is equivalent to purely $SPP$ (Theorem 4).We note that purely $SPP$ and $SPP$ coincide (vacuously) when $n \leq 3$ and that generally the totally $SPP$ matrices are contained in the purely $SPP$ matrices (vacuously). Lastly, but importantly, observe that, if $A$ is totally (purely) $SPP$, then so is any normalization of $A$. Thus, in trying to show that a totally (purely) $SPP$ matrix is $IM$, by positive diagonal equivalence, we may, and do, assume that $A$ is normalized.

Below is our first key result. We follow traditional submatrix notation: for $\phi \neq \alpha, \beta \subseteq N$, $A[\alpha, \beta]$ $(A(\alpha, \beta))$ denotes the submatrix of $A$ with rows indexed by $\alpha$ $(\alpha^c)$ and columns by $\beta$ $(\beta^c)$. For brevity, we denote $A[\alpha, \alpha]$ $(A(\alpha, \alpha))$ by $A[\alpha]$ $(A(\alpha))$.

THEOREM 1. *Every* $IM$ *matrix is purely* $SPP$

*Proof*. Without loss of generality, let $A = (a_{ij})$ be a normalized $IM$ matrix. If $n \leq 3$, then $A$ is purely $SPP$ vacuously. So we may assume that $n \geq 4$. Assume that $a_{ik} = a_{ij}a_{jk}$ for the distinct indices $i, j, k$ of $N$ and let $m \in N - \{i, j, k\}$. Consider the principal submatrix

$$A[\{m, j, k, i\}] = \begin{bmatrix} 1 & a_{mj} & a_{mk} & a_{mi} \\ a_{jm} & 1 & a_{jk} & a_{ji} \\ a_{km} & a_{kj} & 1 & a_{ki} \\ a_{im} & a_{ij} & a_{ik} & 1 \end{bmatrix}$$

of $A$. This submatrix is $IM$ by inheritance. So (via, for instance, the special case of Sylvester's identity for determinants given in [11]) the $k$, $i$ cofactor $c_{ki} = (a_{mk} - a_{mj}a_{jk})(a_{im} - a_{ij}a_{jm}) \leq 0$. Hence, by condition (2), either $a_{mk} = a_{mj}a_{jk}$ or $a_{im} = a_{ij}a_{jm}$. Thus, $A$ is purely $SPP$. ☐

A graph may be constructed on the path product equality triples with an edge corresponding to coincidence of the first two or last two indices. This can be informative about the structure of path product equalities, but all we need here is the following lemma. The important idea is that the occurrence of path product *equalities* ensure that certain submatrices have rank 1 and, moreover, these rank 1 submatrices are large enough to guarantee that certain almost principal minors vanish. We make use of the well-known fact that an $n$-by-$n$ matrix is singular if it contains an $s$-by-$t$ submatrix of rank $r$ such that $s + t \geq n + r + 1$.

LEMMA 1. [illegible] $(i,j,k)$ [illegible] $n$ [illegible] $n$ [illegible] $SPP$ [illegible] $A$ [illegible] $n \geq 4$ [illegible] $\det A(\{k\},\{i\}) = 0$. [illegible] $(k,i)$ [illegible] $A$ [illegible] $p$ $\det A^{(p)}(\{k\},\{i\}) = 0$

[illegible]. Without loss of generality, assume that $A$ is an $n$-by-$n$ normalized purely $SPP$ matrix, $n \geq 4$, and that $(i,j,k)$ is a path product equality for $A$. By permutation similarity, we may assume that $(i,j,k) = (1,2,3)$ so that $a_{13} = a_{12}a_{23}$. Then, since $A$ is purely $SPP$, for each $c \in N - \{1,2,3\}$, either $a_{1c} = a_{12}a_{2c}$ or $a_{c3} = a_{c2}a_{23}$. Without loss of generality, assume that $\{\, c \mid c \in N - \{1,2,3\} \;\text{ and }\; a_{1c} = a_{12}a_{2c}\} \neq \phi$. By permutation similarity, we may assume, again without loss of generality, that, for some $q \in N - \{1,2,3\}$,

(i) $a_{1c} = a_{12}a_{2c}, c = 3, \ldots, q$;

(ii) $a_{1c} \neq a_{12}a_{2c}, c = q+1, \ldots, n$.

Note that, for $c \in \{3, \ldots, q\}$ and $r \in \{q+1, \ldots, n\} \subseteq N - \{1,2,c\}$, (i) implies (by condition (3)) that either $a_{1r} = a_{12}a_{2r}$ or $a_{rc} = a_{r2}a_{2c}$. Hence, by (ii), we have

iii) $a_{rc} = a_{r2}a_{2c}, r = q+1, \ldots, n, c = 3, \ldots, q$.

If $q = n$ and $B = A[\{1,2\},\{2,\ldots,q\}]$, then (i) implies that $[a_{12}\, a_{13}\, \ldots\, a_{1q}]$, the first row of $B$, is $a_{12}$ times $[1\, a_{23}\, a_{24}\, \ldots\, a_{2q}]$, the second row of $B$. Thus, $B$ is a 2-by-$(n-1)$ rank 1 submatrix of $A(\{3\},\{1\})$. Since $2+(n-1) = n+1 \geq n+1 = (n-1)+1+1$, $A(\{3\},\{1\})$ is singular.

On the other hand, if $3 \leq q < n$, let $C = A[\{1,2,q+1,\ldots,n\},\{2,\ldots,q\}]$. Then, (i) implies that $[a_{12}\, a_{13}\, \ldots\, a_{1q}]$, the first row of $C$, is $a_{12}$ times $[1\, a_{23}\, a_{24}\, \ldots\, a_{2q}]$, the second row of $C$. Now consider $[a_{r2}\, a_{r3}\, \ldots\, a_{rq}]$, the $r$th row of $C$, $r = q+1, \ldots, n$. It follows from (iii) that the $r$th row of $C$ is $a_{r2}$ times the second row of $C$, $r = q+1, \ldots, n$. Hence, $C$ is an $(n-q+2)$-by-$(q-1)$ rank 1 submatrix of $A(\{3\},\{1\})$. Since $(n-q+2)+(q-1) = n+1 \geq n+1 = (n-1)+1+1$, $A(\{3\},\{1\})$ is singular in this case also. Thus, $\det A(\{3\},\{1\}) = 0$ in either case, completing the proof of the first part.

The second part follows by the same argument, since condition (3) holds for $A$ if and only if it holds for $A^{(p)}$ for any real number $p$. $\square$

Now using Theorem 1, we have the following.

COROLLARY 1. [illegible] $(i,j,k)$ [illegible] $n$ [illegible] $n$ $IM$ [illegible] $A$ $n \geq 4$ [illegible] $(A^{-1})_{ki} = 0$

THEOREM 2. [illegible] $A$ [illegible] $n$ [illegible] $n$ $SPP$ [illegible] $P > 0$ [illegible] $\det A^{(p)} > 0$ [illegible] $p > P$

[illegible]. Without loss of generality, let $A$ be an $n$-by-$n$ normalized $SPP$ matrix and let $\max_{i \neq j} a_{ij} = M < 1$. Denote the set of permutations of $N$ by $S_n$ and the identity permutation by [illegible]. Then,

$$\begin{aligned}
det\, A^{(p)} &= \sum_{\tau \in S_n} sgn(\tau) a^p_{1,\tau(1)} a^p_{2,\tau(2)} \cdots a^p_{n,\tau(n)} \\
&= 1 + \sum_{\substack{\tau \in S_n \\ \tau \neq id}} sgn(\tau) a^p_{1,\tau(1)} a^p_{2,\tau(2)} \cdots a^p_{n,\tau(n)} \\
&> 1 - \sum_{\substack{\tau \in S_n \\ \tau \neq id}} a^p_{1,\tau(1)} a^p_{2,\tau(2)} \cdots a^p_{n,\tau(n)} \\
&> 1 - (n! - 1)M^p.
\end{aligned}$$

It is clear that there exists $P > 0$ such that for all $p > P$, $1 - (n!-1)M^p > 0$, completing the proof. $\square$

THEOREM 3. [illegible] $A$ [illegible] $n$ [illegible] $n$ [illegible] $SPP$ [illegible] $P > 0$ [illegible] $A^{(p)} \in IM$ [illegible] $p > P$

[illegible]. Without loss of generality, let $A$ be an $n$-by-$n$ normalized, totally $SPP$ matrix. As in the proof of Theorem 2, let $\max_{i \neq j} a_{ij} = M < 1$, let $S_n$ denote the set of permutations of $N$, and let [illegible] denote the identity permutation. Then, it follows from Theorem 2 that there exists $P_1 > 0$ such that, for all $p > P_1$, $\det A^{(p)} > 0$.

Now let $i, j \in N$ with $i \neq j$, and let $N_1 = N - \{i, j\}$. Consider $c_{ij}^{(p)}$, the $i, j$ cofactor of $A^{(p)}$. Without loss of generality, assume that $i < j$. Then, if $B = \begin{bmatrix} A^{(p)}[N_1] & A^{(p)}[N_1, i] \\ A^{(p)}[j, N_1] & a_{ji}^p \end{bmatrix}$,

$$\begin{aligned} c_{ij}^{(p)} &= (-1)^{i+j} \det A^{(p)}(\{i, j\}) \\ &= (-1)^{i+j}(-1)^{n-i-1}(-1)^{n-j} \det B \\ &= (-1) \det B \\ &= -\sum_{\tau \in S_{n-1}} sgn(\tau) b_{1,\tau(1)} b_{2,\tau(2)} \dots b_{n-1,\tau(n-1)} \\ &= -a_{ji}^p - \sum_{\substack{\tau \in S_{n-1} \\ \tau \neq id}} sgn(\tau) b_{1,\tau(1)} b_{2,\tau(2)} \dots b_{n-1,\tau(n-1)} \\ &\leq -a_{ji}^p + \sum_{\substack{\tau \in S_{n-1} \\ \tau \neq id}} b_{1,\tau(1)} b_{2,\tau(2)} \dots b_{n-1,\tau(n-1)}. \end{aligned}$$

Now let

$$\begin{aligned} \Lambda_1 &= \{ \tau \in S_{n-1} \mid \tau \neq id \text{ and } \tau(n-1) = n-1 \} \\ &= \{ \tau \in S_{n-1} \mid \tau \neq id \text{ and } b_{n-1,\tau(n-1)} = a_{ji}^p \} \end{aligned}$$

and

$$\begin{aligned} \Lambda_2 &= \{ \tau \in S_{n-1} \mid \tau(n-1) \neq n-1 \} \\ &= \{ \tau \in S_{n-1} \mid b_{n-1,\tau(n-1)} \neq a_{ji}^p \}, \end{aligned}$$

so that $\{id\}$, $\Lambda_1$, and $\Lambda_2$ form a partition of $S_{n-1}$. Thus,

$$\begin{aligned} \sum_{\substack{\tau \in S_{n-1} \\ \tau \neq id}} b_{1,\tau(1)} b_{2,\tau(2)} \dots b_{n-1,\tau(n-1)} &= \sum_{\tau \in \Lambda_1} b_{1,\tau(1)} b_{2,\tau(2)} \dots b_{n-1,\tau(n-1)} \\ &\quad + \sum_{\tau \in \Lambda_2} b_{1,\tau(1)} b_{2,\tau(2)} \dots b_{n-1,\tau(n-1)}. \end{aligned} \tag{4}$$

Each term of the first summation on the right-hand side of (4) is the product of $b_{n-1,\tau(n-1)} = a_{ji}^p$ and a "nonidentity" term of the expansion of $\det A[N_1]$ (since $\tau \neq id$). Hence, each term in this summation has a factor of the form

$$a_{ji}^p a_{si_1}^p a_{i_1 i_2}^p \dots a_{i_{k-1} i_k}^p a_{i_k s}^p \tag{5}$$

in which $s, i_1, \dots, i_k$ are distinct indices in $N_1$ and $k \geq 1$. Therefore, the cycle product (5) has at least three terms. Since the factors of this term distinct from $a_{ji}^p$ are $< 1$, each term in the first summation is $< a_{ji}^p M^{2p}$, and there are $|S_{n-2}| - 1 = (n-2)! - 1$ such terms.

On the other hand, each term of the second summation on the right-hand side of (4) has a factor of the form

$$a^p_{ji_1} a^p_{i_1 i_2} a^p_{i_2 i_3} \cdots a^p_{i_{k-1} i_k} a^p_{i_k i} \tag{6}$$

in which $i_1, \ldots, i_k$ are distinct indices in $N_1$ with $k \geq 1$. Hence, the path product (6) has at least two terms. Let $m_{ji}$ denote the maximum $(j,i)$ path product given by (6). Therefore, each term in the second summation is $\leq m_{ji}$, and there are $(n-1)! - (n-2)! = (n-2)((n-2)!)$ such terms. Notice that, since $A$ is totally $SPP$, $m_{ji} < a_{ji}$. Thus,

$$\begin{aligned} c^{(p)}_{ij} &\leq -a^p_{ji} + ((n-2)! - 1)\, a^p_{ji} M^{2p} + (n-2)((n-2)!)\, m^p_{ji} \\ &= -a^p_{ji} \left(1 - ((n-2)! - 1)\, M^{2p} - (n-2)((n-2)!)\, (\tfrac{m_{ji}}{a_{ji}})^p\right). \end{aligned} \tag{7}$$

Since $M < 1$ and $m_{ji} < a_{ji}$, there exists $P_{ij} > 0$ such that, for all $p > P_{ij}$, $c^{(p)}_{ij} \leq 0$. Let $P_2 = \max_{i \neq j} P_{ij}$. Then, for all $p > \max(P_1, P_2) = P$, the inverse of $A^{(p)}$ has non-positive off-diagonal entries and, hence, $A^{(p)} \in IM$, completing the proof. $\square$

Our main result characterizes $EIM$ matrices and, in a certain sense, provides a converse to the statement that $IM$ implies (some kind of) $SPP$.

THEOREM 4. [illegible] $A$ [illegible] $n$ [illegible] $n$ [illegible] $A$ [illegible] $SPP$ [illegible] $A \in EIM$

[illegible]. We observe that, for either of the two properties in question, purely $SPP$ or $EIM$, we may assume that $A$ is an $n$-by-$n$ normalized $SPP$ matrix. Necessity of the condition purely $SPP$ then follows from Theorem 1 and the previously noted fact that the path product inequalities (and equalities) are preserved for any positive Hadamard power $p$.

For sufficiency, let $A$ be an $n$-by-$n$ normalized purely $SPP$ matrix. It follows from Theorem 2 that there exists $P_1 > 0$ such that $\det A^{(p)} > 0$ for all $p > P_1$. So we are left to show that, for some $P > 0$, the off-diagonal entries of $(A^{(p)})^{-1}$ are nonpositive for all $p > P$. To this end, let $i, j \in N$ with $i \neq j$, and $N_1 = N - \{i, j\}$. If $a_{ji} = a_{jk} a_{ki}$ for some $k \in N_1$, then it follows from Lemma 1 that $c^{(p)}_{ij}$, the $i,j$ cofactor of $A^{(p)}$, vanishes for all positive $p$. Let $P_{ij} = 1$ in this case. On the other hand, suppose that

$$a_{ji} > a_{jk} a_{ki} \tag{8}$$

for all $k \in N_1$. Following the proof of Theorem 3, let $\max_{i \neq j} a_{ij} = M < 1$ and let $m_{ji}$ denote the maximum $(j,i)$ path product given by (6). That $m_{ji} < a_{ji}$ follows from (8). Hence, (7) implies that there is a positive constant $P_{ij}$ such that, for all $p > P_{ij}$, $c^{(p)}_{ij}$, the $i,j$ cofactor of $A^{(p)}$, is $\leq 0$. Letting $P_2 = \max_{i \neq j} P_{ij}$ and $P = \max(P_1, P_2)$, we see that for all $p > P$, $A^{(p)}$ is invertible, and its inverse has nonpositive off-diagonal entries; that is, $A^{(p)}$ is $IM$. So $A$ is $EIM$, completing the proof. $\square$

[illegible] Suppose that we have a totally (purely) $SPP$ matrix that is not $IM$. Since the totally (purely) $SPP$ matrices are closed under any positive Hadamard power, extraction of a small enough Hadamard root will produce a totally (purely) $SPP$ matrix in which $P$ must be arbitrarily large, while raising the matrix to a large enough Hadamard power will produce a totally (purely) $SPP$ matrix in which $P$ may be taken to be an arbitrarily small positive number. In fact, $P$ can be 0.

Immediately from Theorem 1, we have the following result.

COROLLARY 2. [illegible] $A$ [illegible] $IM$ [illegible] $P > 0$ [illegible] $A^{(p)} \in IM$ [illegible] $p > P$

## REFERENCES

[1] J. P. Benzécri, ed., *L'Analyse des Données, Vol.* I: *La Taxinomie*, Dunod, Paris, 1973.

[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, San Diego, 1979.

[3] S. Chen, *A property concerning the Hadamard powers of inverse M-matrices*, Linear Algebra Appl., 381 (2004), pp. 53–60.

[4] C. Dellacherie, S. Martínez, and J. San Martín, *Ultrametric matrices and induced Markov chains*, Adv. Appl. Math., 17 (1996), pp. 169–183.

[5] C. Dellacherie, S. Martínez, and J. San Martín, *Description of the sub-Markov kernel associated to generalized ultrametric matrices: An algorithmic approach*, Linear Algebra Appl., 318 (2000), pp. 1–21.

[6] D. Capocacia, M. Cassandro, and P. Picco, *On the existence of thermodynamics for the generalized random energy model*, J. Statist. Physics, 46 (1987), pp. 493–505.

[7] M. Fiedler, *Special ultrametric matrices and graphs*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 106–6113.

[8] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, New York, 1991.

[9] C. R. Johnson, *Inverse M-matrices*, Linear Algebra Appl., 47 (1982), pp. 195–216.

[10] C. R. Johnson and R. L. Smith, *Path product matrices*, Linear Multilinear Algebra, 46 (1999), pp. 177–191.

[11] C. R. Johnson and R. L. Smith, *Almost principal minors of inverse M-matrices*, Linear Algebra Appl., 337 (2001), pp. 253–265.

[12] I. Koltracht and M. Neumann, *On the Inverse M-matrix problem for real symmetric positive-definite Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 310–320.

[13] S. Martínez, G. Michon, and J. San Martín, *Inverse of strictly ultrametric matrices are of Stieltjes type*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 98–106.

[14] S. Martínez, J. San Martín, and X.-D. Zhang, *A new class of inverse M-matrices of tree-like type*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1136–1148.

[15] R. A. Willoughby, *The inverse M-matrix problem*, Linear Algebra Appl., 18 (1977), pp. 75–94.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 377–395

# A SUBSPACE-BASED METHOD FOR SOLVING LAGRANGE–SYLVESTER INTERPOLATION PROBLEMS*

HÜSEYIN AKÇAY† AND SEMIHA TÜRKAY†

**Abstract.** In this paper, we study the Lagrange–Sylvester interpolation of rational matrix functions which are analytic at infinity, and propose a new interpolation algorithm based on the recent subspace-based identification methods. The proposed algorithm is numerically efficient and delivers a minimal interpolant in state-space form. The solvability condition for the subspace-based algorithm is particularly simple and depends only on the total multiplicity of the interpolation nodes. As an application, we consider subspace-based system identification with interpolation constraints, which arises, for example, in the identification of continuous-time systems with a given relative degree.

**Key words.** rational interpolation, Lagrange–Sylvester, identification, subspace-based

**AMS subject classifications.** 93A30, 30E05, 65D05

**DOI.** 10.1137/050622171

## 1. Introduction.

Many problems in control, circuit theory, and signal processing can be reduced to the solution of matrix rational interpolation problems which have been widely studied (see, for example, [14, 15, 2, 19, 21, 3, 4, 1, 5, 6, 7, 31, 30, 11, 10] and the references therein). Applications arise, for example, in robust controller synthesis [19, 21], in the $Q$-parameterization of stabilizing controllers for unstable plants [18], in the problem of model validation [32], in circuit theory [34], in spectral estimation [12], and in adaptive filtering and control [31, 26].

In the simplest form, given complex numbers $z_k$ and $w_k$ for $k = 1, \ldots, N$, an interpolation problem asks for scalar rational functions $G(z)$ which meet the interpolation conditions

$$G(z_k) = w_k, \qquad k = 1, \ldots, N.$$

The interpolants can further be required to have minimal complexity in terms of their McMillan degree. Let **R** and **C** denote the fields of the real and complex numbers, respectively. An extension of this problem to the matrix case is as follows.

[illegible] a subset $\vartheta \subset \mathbf{C}$, points $z_1, \ldots, z_L$ in $\vartheta$, rational $1 \times p$ row vector functions $v_1(z), \ldots, v_L(z)$ with $v_k(z_k) \neq 0$ for all $k$, and rational $1 \times m$ row vectors $w_1(z), \ldots, w_L(z)$.

[illegible] (at least one or all) $p \times m$ rational matrix functions $G(z)$ with no poles in $\vartheta$ which satisfy the [illegible] interpolation conditions

$$\frac{d^j}{dz^j}\{v_k(z)G(z)\}\bigg|_{z=z_k} = \frac{d^j}{dz^j} w_k(z)\bigg|_{z=z_k} \tag{1.1}$$

for $0 \leq j \leq N_k$, $1 \leq k \leq L$.

This problem is known as the tangential [illegible] rational interpolation problem. One approach to finding a solution is to reduce the problem to a system of

---

*Received by the editors January 7, 2005; accepted for publication (in revised form) by P. Benner October 26, 2006; published electronically March 19, 2007.
http://www.siam.org/journals/simax/29-2/62217.html

†Department of Electrical and Electronics Engineering, Anadolu University, 26470 Eskişehir, Turkey (huakcay@anadolu.edu.tr, semihaturkay@anadolu.edu.tr).

independent scalar problems, which is not interesting from the viewpoint of matrix interpolation theory. In addition, a minimal realization can be obtained only after the elimination of unobservable or/and uncontrollable modes. The contour integral version of this problem is treated in the comprehensive work [6]. The [illegible] or [illegible] version is studied, for example, in [14, 15, 7, 4]. Related problems are the nonhomogenous interpolation problem with metric constraints, as in the various types of Nevanlinna–Pick interpolation and its generalizations [10, 20], and the partial realization problem, that is, finding a rational matrix function analytic at infinity of the smallest possible McMillan degree with prescribed values of itself and a few of its derivatives at infinity [17, 1, 6, 27, 28]. Further applications of interpolation theory to control and systems theory and estimation are presented in [6, 13, 29].

Prior work on the unconstrained tangential interpolation problem has been largely carried out by Ball, Gohberg, and Rodman [6, 7]. The solvability issues of the interpolation problem, i.e., the existence and the uniqueness of the solutions, have been analyzed in [8] by using a [illegible] framework. A more direct algebraic approach in [11] shows that solving a tangential interpolation problem is equivalent to solving a matrix Padé approximation problem with Taylor coefficients obeying a set of linear constraints. In [1, 2, 3, 4], the tangential interpolation problem above was studied using a tool called the Löwner matrix. In [4], the problem of finding admissible degrees of complexity of the solutions to the above interpolation problem, that is, finding all positive integers $n$ for which there exits an interpolant with McMillan degree $n$, and the problem of parameterizing all solutions for a given admissible degree of complexity were investigated. Clearly, the solutions of minimal complexity are of special interest.

The main result in [7] states that the family of rational matrix functions satisfying (1.1) can be parameterized in terms of a certain [illegible]. First, the interpolation data is translated into a so-called [illegible] that describes the zero structure of a $(p+m) \times (p+m)$ [illegible]. The computation of the resolvent matrix requires that the solution of a particular Sylvester equation be invertible. The details can be found in [6]. In [11], a recursive method for computing the resolvent matrix as a product of elementary first-order rational matrix functions is presented. This scheme allows recursive updating of the resolvent matrix whenever a new interpolation point is added to the input data. In the special case when the resolvent matrix is in [illegible] form, it is possible to extract the admissible degrees of complexity as well as the minimal degree of complexity from the linear fractional parameterization formula. The resolvent matrix obtained by an unconstrained algorithm can be transformed into column-reduced form via a sequence of elementary unimodular transformations [16]. A detailed algorithm for the construction of a column-reduced rational matrix function from a given [illegible] is given in [9]. This algorithm is not recursive, whereas in [11] a column-reduced transfer function is recursively obtained.

In this paper, we present a numerically efficient algorithm for solving the unconstrained tangential interpolation problem formulated above. This algorithm is inspired by the recent work on the frequency domain subspace-based identification [23, 24, 25, 33]. The solvability conditions for the proposed algorithm are simple, and depend only on the total multiplicities of the interpolation points. The resulting interpolating function is in the minimal state-space form. To this date, interpolation properties of the subspace-based methods have not been investigated in the generality of this paper. Only in [24] was an interpolation result obtained for uniformly spaced data on the unit circle of the complex plane. The problem of curve fitting is also closely related to the interpolation problem. The use of the frequency domain subspace-based methods for curve fitting is briefly described in [22].

Let us reformulate the tangential interpolation problem described above in terms of system properties. More precisely, let us consider a multi-input/multi-output, linear-time invariant, discrete-time system represented by the state-space equations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k), \end{aligned} \tag{1.2}$$

where $x(k) \in \mathbf{R}^n$ is the state and $u(k) \in \mathbf{R}^m$ and $y(k) \in \mathbf{R}^p$ are, respectively, the input and the output of the system. The transfer function of the system (1.2) denoted by $G(z)$ is computed as

$$G(z) = D + C(zI_n - A)^{-1}B, \tag{1.3}$$

where $I_n$ is the $n \times n$ identity matrix. We assume that that the system (1.2) is stable and the pairs $(A, B)$ and $(C, A)$ are [illegible] and [illegible], respectively. The stability of (1.2) means that $G(z)$ is a [illegible] matrix that is analytic and bounded in the region $\vartheta = \{z \in \mathbf{C} : |z| \geq 1\}$, and both the controllability and the observability of the pairs $(A, B)$ and $(C, A)$ mean that the quadruplet $(A, B, C, D)$ is a [illegible] of $G(z)$.

The interpolation problem studied in this paper can be stated as follows.

[illegible] noise-free samples of $G(z)$ and its derivatives at $L$ distinct points $z_k \in \vartheta$,

$$\left. \frac{d^j}{dz^j} G(z) \right|_{z=z_k} = w_{kj}, \qquad j = 0, 1, \ldots, N_k, \quad k = 1, 2, \ldots, L. \tag{1.4}$$

[illegible] a quadruplet $(\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$ that is a minimal realization of $G(z)$.

Clearly, (1.4) is a special case of (1.1) with suitably selected left vectors $v_k(z)$ and nodes $z_k$. A subspace-based algorithm handling the tangential-type constraints (1.1) as well can be derived along the same lines of the proposed algorithm. The minimality and the uniqueness of the interpolant are the parts of the problem formulation. What is left unanswered is a condition on the number of the interpolation nodes, counting multiplicities. It is also clear that, if it exists, the subspace-based solution is a minimal interpolating function in the set of all possible solutions.

The proposed interpolation scheme is particularly useful when the samples of $G(z)$ and its derivatives are corrupted by noise and the amount of data is large with respect to $n$. In the noisy case, most interpolation schemes deliver state-space realizations with McMillan degrees tending to infinity as the amount of data grows unboundedly; thus such schemes are sensitive to inaccuracies in the interpolation data. Since our algorithm is subspace-based, it inherits robustness properties of the subspace-based identification algorithms. In particular, there is no need for explicit model parameterization, and this algorithm is computationally efficient since it uses numerically robust QR factorization and the singular value decomposition. In the paper, we also consider subspace-based system identification with interpolation constraints.

Note that a given interpolation problem on the right half complex plane can be converted to an interpolation problem on the unit disk by using the Möbius transformation:

$$s = \psi(z) \triangleq \lambda \frac{z-1}{z+1} \qquad (\lambda > 0). \tag{1.5}$$

We omit the details.

**2. Subspace-based interpolation algorithm.** We begin by taking the $z$-transform of (1.2),

$$\begin{aligned} zX(z) &= AX(z) + BU(z), \\ Y(z) &= CX(z) + DU(z), \end{aligned} \tag{2.1}$$

assuming $x(0) = 0$, where $X(z)$, $Y(z)$, and $U(z)$ denote respectively the $z$-transforms of $x(k)$, $y(k)$, and $u(k)$ defined by

$$U(z) \triangleq \sum_{k=0}^{\infty} u(k)\, z^{-k}. \tag{2.2}$$

Let $X_j(x)$ be the resulting state $z$-transform when

$$u(k) = \begin{cases} e_j, & k = 0, \\ 0, & \text{otherwise}, \end{cases}$$

where $e_j$ denotes the unit vector in $\mathbf{R}^m$ with 1 on the $j$th position and 0 elsewhere. By defining the compound state $z$-transform matrix,

$$X_{\mathrm{C}}(z) \triangleq \left[X_1(z) \;\; X_2(z) \;\; \cdots \;\; X_m(z)\right], \tag{2.3}$$

$G(z)$ can implicitly be described as

$$G(z) = CX_{\mathrm{C}}(z) + D \tag{2.4}$$

with

$$zX_{\mathrm{C}}(z) = AX_{\mathrm{C}}(z) + B. \tag{2.5}$$

By recursive use of (2.5), we obtain the relation

$$z^k X_{\mathrm{C}}(z) = A^k X_{\mathrm{C}}(z) + \sum_{j=0}^{k-1} A^{k-1-j} B z^j, \qquad k \geq 1. \tag{2.6}$$

Multiplying both sides of (2.6) with $C$ and using (2.4), we get

$$z^k G(z) = CA^k X_{\mathrm{C}}(z) + Dz^k + \sum_{j=0}^{k-1} CA^{k-1-j} B z^j, \qquad k \geq 1. \tag{2.7}$$

Now, recall that the impulse response coefficients of $G(z)$ are given by

$$g_k = \begin{cases} D, & k = 0, \\ CA^{k-1}B, & k \geq 1. \end{cases} \tag{2.8}$$

Thus, from (2.4), (2.7), and (2.8),

$$z^k G(z) = CA^k X_{\mathrm{C}}(z) + \sum_{j=0}^{k} g_{k-j}\, z^j, \qquad k \geq 0. \tag{2.9}$$

Hence from (2.9),

$$\begin{bmatrix} G(z) \\ zG(z) \\ \vdots \\ z^{q-1}G(z) \end{bmatrix} = \mathcal{O}_q X_{\mathrm{C}}(z) + \Gamma_q \begin{bmatrix} I_m \\ zI_m \\ \vdots \\ z^{q-1}I_m \end{bmatrix}, \tag{2.10}$$

where

$$\mathcal{O}_q \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{q-1} \end{bmatrix}, \tag{2.11}$$

$$\Gamma_q \triangleq \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{q-1} & g_{q-2} & \cdots & g_0 \end{bmatrix}. \tag{2.12}$$

For later use, let us write (2.10) in a compact form. The matrix $\mathcal{O}_q$ is known as the [illegible] and has full rank $n$ if $(A, C)$ is an observable pair and $q \geq n$. We define the [illegible] of two matrices $E \in \mathbf{C}^{m\times n}$ and $F \in \mathbf{C}^{p\times q}$ by

$$E \otimes F \triangleq \begin{bmatrix} E_{11}F & E_{12}F & \cdots & E_{1n}B \\ E_{21}F & E_{22}F & \cdots & E_{2n}F \\ \vdots & \vdots & \ddots & \vdots \\ E_{m1}F & E_{m2}F & \cdots & E_{mn}F \end{bmatrix} \in \mathbf{C}^{mp\times nq}. \tag{2.13}$$

Let

$$\mathcal{Z}_q(z) \triangleq \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{q-1} \end{bmatrix}, \tag{2.14}$$

$$\mathcal{J}_{q,2} \triangleq \begin{bmatrix} 0 & & \cdots & & 0 \\ 1 & 0 & & & \\ 0 & 1 & 0 & & \\ \vdots & & & \ddots & \vdots \\ 0 & & \cdots & 1 & 0 \end{bmatrix} \in \mathrm{R}^{q\times q}. \tag{2.15}$$

By a slight abuse of notation, let $\mathcal{J}_{q,1}$ denote the $q \times q$ identity matrix $I_q$. Observe that $\mathcal{J}_{q,2}$ is obtained by shifting the elements of $\mathcal{J}_{q,1}$ one row down and filling its first row with zeros. Let $\mathcal{J}_{q,j}$ denote the matrix obtained by $j-1$ repeated applications of this process to $\mathcal{J}_{q,1}$ and $J_{q,2}^0 = I_q$. Note the following relations:

$$\mathcal{J}_{q,j} = \begin{cases} \mathcal{J}_{q,2}^{j-1}, & j \leq q \\ 0, & j > q. \end{cases} \tag{2.16}$$

Thus, the lower triangular block Toeplitz matrix in (2.12) can be written as

$$\Gamma_q = \sum_{j=0}^{q-1} \mathcal{J}_{q,1+j} \otimes g_j . \tag{2.17}$$

Hence, from (2.11)–(2.17) we arrive at the following compact expression for (2.10):

$$\mathcal{Z}_q(z) \otimes G(z) = \mathcal{O}_q X_{\mathrm{C}}(z) + \sum_{j=0}^{q-1} [\mathcal{J}_{q,2}^{j} \otimes g_j] \; [\mathcal{Z}_q(z) \otimes I_m] \, . \tag{2.18}$$

This equation forms the basis of the frequency domain subspace-based identification algorithms [24, 23]. In subspace-based identification algorithms, $\mathcal{Z}_q(z) \otimes G(z)$ and the right-hand side of (2.18) are evaluated at a set of distinct points on the unit circle and then stacked into columns of long matrices. This procedure yields a matrix equation [illegible] in $\mathcal{O}_q$. From this equation, the range space of $\mathcal{O}_q$ is recovered by a projection. Once the observability range space is recovered, a realization of $G(z)$ is derived in a routine manner. We will adapt the same strategy.

First, we differentiate both sides of (2.18) $l$ times with respect to $z$:

$$\begin{aligned} \frac{d^l}{dz^l} H_q(z) &= \sum_{j=0}^{l} \begin{pmatrix} l \\ j \end{pmatrix} \frac{d^j}{dz^j} \mathcal{Z}_q(z) \otimes \frac{d^{l-j}}{dz^{l-j}} G(z) \\ &= \mathcal{O}_q \, \frac{d^l}{dz^l} X_{\mathrm{C}}(z) + \sum_{j=0}^{q-1} [\mathcal{J}_{q,2}^{j} \otimes g_j] \left[ \frac{d^l}{dz^l} \mathcal{Z}_q(z) \otimes I_m \right], \qquad l \geq 0, \end{aligned} \tag{2.19}$$

where

$$H_q(z) \triangleq \mathcal{Z}_q(z) \otimes G(z). \tag{2.20}$$

Then, we augment $H_q(z_k)$ and the first $N_k$ derivatives of $H_q(z)$ at $z_k$ in a data matrix:

$$\mathcal{H}_k \triangleq \left[ H_q(z) \;\; \frac{d}{dz} H_q(z) \;\; \cdots \;\; \frac{d^{N_k}}{dz^{N_k}} H_q(z) \right]_{z=z_k} , \qquad k = 1, \ldots, L. \tag{2.21}$$

Using the right-hand side of the first equality in (2.19), let us derive a compact expression for $\mathcal{H}_k$ in terms of the elementary matrices

$$\mathcal{D}_k \triangleq \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ & 0 & 2 & & \\ & & 0 & \cdots & \\ \vdots & & & \ddots & N_k \\ 0 & & & \cdots & 0 \end{bmatrix} \in \mathbf{R}^{(N_k+1) \times (N_k+1)} \tag{2.22}$$

and

$$\mathcal{W}_k \triangleq \left[ \mathcal{Z}_q(z) \;\; \frac{d}{dz} \mathcal{Z}_q(z) \;\; \cdots \;\; \frac{d^{N_k}}{dz^{N_k}} \mathcal{Z}_q(z) \right]_{z=z_k} , \qquad k = 1, \ldots, L, \tag{2.23}$$

as follows:

$$\begin{aligned}
\mathcal{H}_k = {} & \left[\mathcal{Z}_q(z)\ \frac{\mathrm{d}}{dz}\mathcal{Z}_q(z)\ \frac{d^2}{dz^2}\mathcal{Z}_q(z)\ \cdots\ \frac{d^{N_k}}{dz^{N_k}}\mathcal{Z}_q(z)\right]_{z=z_k} \otimes G(z_k) \\
& + \left[0\ \ \mathcal{Z}_q(z)\ \ 2\frac{d}{\mathrm{d}z}\mathcal{Z}_q(z)\ \ \cdots\ \ \begin{pmatrix} N_k \\ 1 \end{pmatrix} \frac{d^{N_k-1}}{dz^{N_k-1}}\mathcal{Z}_q(z)\right]_{z=z_k} \otimes \frac{d}{dz}G(z_k) \\
& + \left[0\ \ 0\ \ \mathcal{Z}_q(z)\ \ \cdots\ \ \begin{pmatrix} N_k \\ 2 \end{pmatrix} \frac{d^{N_k-2}}{dz^{N_k-2}}\mathcal{Z}_q(z)\right]_{z=z_k} \otimes \frac{d^2}{dz^2}G(z_k) + \cdots \\
& + \left[0\ \ 0\ \ 0\ \ \cdots\ \ \mathcal{Z}_q(z)\right]_{z=z_k} \otimes \frac{d^{N_k}}{dz^{N_k}}G(z_k) \\
= {} & \mathcal{W}_k \otimes G(z_k) + [\mathcal{W}_k\mathcal{D}_k] \otimes \frac{d}{dz}G(z_k) + \frac{1}{2!}\,[\mathcal{W}_k\mathcal{D}_k^2] \otimes \frac{d^2}{dz^2}G(z_k) \\
& + \frac{1}{N_k!}\,[\mathcal{W}_k\mathcal{D}_k^{N_k}] \otimes \frac{d^{N_k}}{dz^{N_k}}G(z_k).
\end{aligned}$$

Note that $\mathcal{D}_k^j = 0$ for all $j > N_k$. Hence,

$$\mathcal{H}_k = \sum_{j=0}^{N_k} \frac{1}{j!}\,[\mathcal{W}_k\,\mathcal{D}_k^j] \otimes w_{kj}, \qquad k = 1, \ldots, L. \tag{2.24}$$

It remains to compute the derivatives of $\mathcal{Z}_q(z)$. To this end, let

$$\mathcal{T}_q \triangleq \begin{bmatrix} 0! & 0 & \cdots & 0 \\ 0 & 1! & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (q-1)! \end{bmatrix} \in \mathbf{R}^{q\times q}. \tag{2.25}$$

Then, it is easy to verify that

$$\frac{d^l}{dz^l}\mathcal{Z}_q(z) = \mathcal{T}_q\mathcal{J}_{q,2}^l\mathcal{T}_q^{-1}\mathcal{Z}_q(z), \qquad l \geq 0. \tag{2.26}$$

Hence from (2.23) and (2.26),

$$\mathcal{W}_k = \mathcal{T}_q\left[I_q\ \ \mathcal{J}_{q,2}\ \ \cdots\ \ \mathcal{J}_{q,2}^{N_k}\right]\left[I_{N_k+1} \otimes \mathcal{T}_q^{-1}\mathcal{Z}_q(z_k)\right], \qquad k = 1, \ldots, L. \tag{2.27}$$

An alternative compact expression for $\mathcal{H}_k$ is obtained by evaluating the right-hand side of the second equality in (2.19) for $l = 0, \ldots, N_k$, $k = 1, \ldots, L$, and augmenting the similar terms in compound matrices as follows:

$$\mathcal{H}_k = \mathcal{O}_q\,\mathcal{X}_k + \sum_{j=0}^{q-1}[\mathcal{J}_{q,2}^j \otimes g_j]\,[\mathcal{W}_k \otimes I_m], \qquad k = 1, \ldots, L, \tag{2.28}$$

where

$$\mathcal{X}_k \triangleq \left[X_{\mathrm{C}}(z)\ \ \frac{d}{dz}X_{\mathrm{C}}(z)\ \ \cdots\ \ \frac{d^{N_k}}{dz^{N_k}}X_{\mathrm{C}}(z)\right]_{z=z_k}, \qquad k = 1, \ldots, L. \tag{2.29}$$

Now, we collect $\mathcal{H}_k$, $\mathcal{X}_k$, and $\mathcal{W}_k$, $k = 1, \ldots, L$, in the compound matrices

$$\mathcal{H} \triangleq [\mathcal{H}_1\ \mathcal{H}_2\ \cdots\ \mathcal{H}_L]\,, \tag{2.30}$$

$$\mathcal{X} \triangleq [\mathcal{X}_1\ \mathcal{X}_2\ \cdots\ \mathcal{X}_L]\,, \tag{2.31}$$

$$\mathcal{W} \triangleq [\mathcal{W}_1\ \mathcal{W}_2\ \cdots\ \mathcal{W}_L]\,. \tag{2.32}$$

Hence,

$$\mathcal{H} = \mathcal{O}_q\,\mathcal{X} + \sum_{j=0}^{q-1}[\mathcal{J}_{q,2}^{j} \otimes g_j]\,[\mathcal{W} \otimes I_m], \tag{2.33}$$

where $\mathcal{H}$ and $\mathcal{W}$ are computed from the problem data $\{z_k, \{w_{kj}\}_{j=0}^{N_k}\}_{k=1}^{L}$ by the formulae (2.30), (2.32), (2.27), (2.24), (2.22), (2.14), (2.15), (2.25). This completes the first stage of our subspace-based interpolation algorithm. Observe that $\mathcal{H}$ is affine in $\mathcal{O}_q$ as advertised.

Since $\mathcal{O}_q$ is a real matrix and we are interested in the real range space, we can convert (2.33) into a relation involving only real valued matrices:

$$\widehat{\mathcal{H}} = \mathcal{O}_q\,\widehat{\mathcal{X}} + \sum_{j=0}^{q-1}[\mathcal{J}_{q,2}^{j} \otimes g_j]\,\mathcal{F}, \tag{2.34}$$

where

$$\widehat{\mathcal{H}} \triangleq [\mathrm{Re}\mathcal{H}\ \ \mathrm{Im}\mathcal{H}]\,, \tag{2.35}$$

$$\mathcal{F} \triangleq [\mathrm{Re}\mathcal{W}\ \ \mathrm{Im}\mathcal{W}] \otimes I_m, \tag{2.36}$$

$$\widehat{\mathcal{X}} \triangleq [\mathrm{Re}\mathcal{X}\ \ \mathrm{Im}\mathcal{X}]\,. \tag{2.37}$$

Let $z^*$ denote the complex conjugate of $z$. When $z_k \in \mathbf{R}$, from (2.14) we have $\mathcal{Z}_q(z_k) \in \mathbf{R}^q$. This, by (2.27), implies that $\mathcal{W}_k \in \mathbf{R}^{q\times(N_k+1)}$. From (2.5),

$$X_{\mathrm{C}}(z) = (zI_n - A)^{-1}B. \tag{2.38}$$

Then, from (2.4), (2.38), and (2.29), it follows that $\mathcal{X}_k \in \mathbf{R}^{n\times m(N_k+1)}$ and, for all $j = 0, \ldots, N_k$, $w_{kj} \in \mathbf{R}^{p\times m}$ whenever $z_k \in \mathbf{R}$. Thus, whenever $z_k \in \mathbf{R}$ from (2.24) we have $\mathcal{H}_k \in \mathbf{R}^{pq\times m(N_k+1)}$. Hence, the imaginary parts of $\mathcal{H}_k$, $\mathcal{F}$, and $\mathcal{X}_k$ are all zero, and they need not be included in (2.35)–(2.37) if $z_k \in \mathbf{R}$; without loss of generality, we will assume this in what follows. Let

$$N \triangleq \sum_{k:z_k\in\mathbf{R}}(N_k+1) + \sum_{k:z_k\in\mathbf{C}-\mathbf{R}} 2(N_k+1). \tag{2.39}$$

Then, $\widehat{\mathcal{H}} \in \mathbf{R}^{pq\times mN}$, $\mathcal{F} \in \mathbf{R}^{mq\times mN}$, and $\widehat{\mathcal{X}} \in \mathbf{R}^{n\times mN}$.

**2.1. Projection onto the observability range space.** Let $\mathcal{F}^{\perp}$ be the projection matrix onto the null space of $\mathcal{F}$ given by

$$\mathcal{F}^{\perp} \triangleq I_{mN} - \mathcal{F}^T(\mathcal{F}\mathcal{F}^T)^{-1}\mathcal{F}, \tag{2.40}$$

where $\mathcal{F}^T$ denotes the transpose of $\mathcal{F}$. The summand in (2.34) is cancelled for all $j$ when multiplied from right by $\mathcal{F}^{\perp}$. Thus,

$$\widehat{\mathcal{H}}\mathcal{F}^{\perp} = \mathcal{O}_q\,\widehat{\mathcal{X}}\mathcal{F}^{\perp}. \tag{2.41}$$

A numerically efficient way of forming $\widehat{\mathcal{H}}\mathcal{F}^{\perp}$ is to use the QR-factorization

$$\begin{bmatrix} \mathcal{F} \\ \widehat{\mathcal{H}} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}. \tag{2.42}$$

A simple derivation yields

$$\widehat{\mathcal{H}}\mathcal{F}^{\perp} = R_{22}Q_2^T, \tag{2.43}$$

and it suffices to use $R_{22} \in \mathbf{R}^{pq\times m(N-q)}$ in the extraction of the observability range space since $Q_2^T$ is a matrix of full rank.

The range space of $\widehat{\mathcal{H}}\mathcal{F}^{\perp}$ equals the range space of $\mathcal{O}_q$ unless rank cancellations occur. A sufficient condition for the range spaces to be equal is that the intersection of the row spaces of $\mathcal{F}$ and $\widehat{\mathcal{X}}$ be empty. In the following, we present sufficient conditions in terms of the data and the system.

LEMMA 2.1. [illegible] $\widehat{\mathcal{X}}$ $\mathcal{F}$ [illegible] $N$ [illegible] (2.37) (2.36) [illegible] (2.39) [illegible] $N \geq q+n$ [illegible] $A$ [illegible] $z_k$ [illegible]

$$\operatorname{rank}\begin{bmatrix} \mathcal{F} \\ \widehat{\mathcal{X}} \end{bmatrix} = qm+n \qquad \Longleftrightarrow \qquad (A,B) \text{ controllable pair.} \tag{2.44}$$

[illegible] The matrix $\begin{bmatrix} \mathcal{W}\otimes I_m \\ \mathcal{X} \end{bmatrix}$ is rank deficient if and only if there exists a row vector

$$[\alpha_0 \ \cdots \ \alpha_{q-1} \ \beta] \neq 0 \tag{2.45}$$

with $\alpha_k^T \in \mathrm{R}^m,\ k=0,\dots,q-1$, and $\beta^T \in \mathrm{R}^n$ such that

$$[\alpha_0 \ \cdots \ \alpha_{q-1} \ \beta]\begin{bmatrix} \mathcal{W}\otimes I_m \\ \mathcal{X} \end{bmatrix} = 0. \tag{2.46}$$

From (2.32), (2.23), and (2.31), (2.29), equation (2.46) holds if and only if

$$[\alpha_0 \ \cdots \ \alpha_{q-1} \ \beta]\frac{d^j}{dz^j}\begin{bmatrix} \mathcal{Z}_q(z)\otimes I_m \\ \mathcal{X}_{\mathrm{C}}(z) \end{bmatrix}_{z=z_k} = 0, \quad 0 \leq j \leq N_k,\ k=1,\dots,L,$$

$$\Updownarrow$$

$$\left.\frac{d^j}{dz^j}E(z)\right|_{z=z_k} = 0, \quad 0 \leq j \leq N_k,\ k=1,\dots,L, \tag{2.47}$$

where

$$E(z) \triangleq \sum_{k=0}^{q-1}\alpha_k z^k + \beta(zI_n - A)^{-1}B.$$

Equation (2.47) implies that for each $k$ the elements of the rational vector $E(z)$ have common zeros at $z_k$ with multiplicity $N_k+1$. Since $E(z)$ is real-rational, $z_k$ is a zero of $E(z)$ if and only if $z_k^*$ is also a zero of $E(z)$. Therefore, $E(z)$ happens to have a total number of $N$ zeros counting multiplicities. However, the elements of $E(z)$ have numerator degrees not exceeding $n+q-1$. Hence, any element of $E(z)$ cannot have $N$ zeros. Thus, $E(z) \equiv 0$. This implies that $\alpha_k = 0$ for all $k$ and $\beta(zI_n - A)^{-1}B \equiv 0$. The latter result follows from the fact that $\beta(zI_n - A)^{-1}B$ is analytic and has a zero at $z=\infty$; hence it is orthogonal to $\sum_{k=0}^{q-1}\alpha_k z^k$. Recall that $(A,B)$ is an uncontrollable pair if and only if it is possible to find a vector $\beta \neq 0$ such that $\beta(zI_n - A)^{-1}B \equiv 0$. Finally, note that $\begin{bmatrix} \mathcal{F} \\ \widehat{\mathcal{X}} \end{bmatrix}$ is rank deficient if and only if $\begin{bmatrix} \mathcal{W} \otimes I_m \\ \mathcal{X} \end{bmatrix}$ is rank deficient. The last assertion is due to the fact that, for any complex matrix $Z$ and real vector $x$,

$$x^T Z = 0 \Longleftrightarrow x\,[\mathrm{Re}Z \ \mathrm{Im}Z] = 0. \qquad \square$$

Since all the eigenvalues of $A$ are inside the unit circle, none of them coincide with any of $z_k$. Thus, by applying Lemma 2.1, we conclude that the two row spaces of $\widehat{\mathcal{X}}$ and $\mathcal{F}$ do not intersect and the range space of $\widehat{\mathcal{H}}\mathcal{F}^\perp$ coincides with the range space of $\mathcal{O}_q$. Then, using the singular value factorization of $\widehat{\mathcal{H}}\mathcal{F}^\perp$,

$$\begin{aligned} \widehat{\mathcal{H}}\mathcal{F}^\perp &= \widehat{U}\widehat{\Sigma}\widehat{V}^T \\ &= \begin{bmatrix} \widehat{U}_s & \widehat{U}_o \end{bmatrix} \begin{bmatrix} \widehat{\Sigma}_s & 0 \\ 0 & \widehat{\Sigma}_o \end{bmatrix} \begin{bmatrix} \widehat{V}_s^T \\ \widehat{V}_o^T \end{bmatrix}, \end{aligned} \tag{2.48}$$

where $\widehat{\Sigma}_s \in \mathbf{R}^{n\times n}$, we determine the system matrices $\widehat{A}$ and $\widehat{C}$ as

$$\begin{aligned} \widehat{A} &= (J_1\widehat{U}_s)^\dagger J_2\widehat{U}_s, \\ \widehat{C} &= J_3\widehat{U}_s, \end{aligned} \tag{2.49}$$

where

$$J_1 = \begin{bmatrix} I_{(q-1)p} & 0_{(q-1)p\times p} \end{bmatrix}, \tag{2.50}$$

$$J_2 = \begin{bmatrix} 0_{(q-1)p\times p} & I_{(q-1)p} \end{bmatrix}, \tag{2.51}$$

$$J_3 = \begin{bmatrix} I_p & 0_{p\times(q-1)p} \end{bmatrix}, \tag{2.52}$$

$0_{i\times j}$ is the $i \times j$ zero matrix, and $X^\dagger = (X^TX)^{-1}X^T$ is the Moore–Penrose pseudoinverse of the full column rank matrix $X$. Provided that $(C, A)$ is an observable pair, the pseudoinverse in (2.49) exists if and only if $q > n$. Therefore, in order to apply the lemma it suffices to let $q = n + 1$. In this case, we have the sole requirement $N > 2n$ with $N$ defined by (2.39). From Lemma 2.1, it follows that $\widehat{A}$ and $\widehat{C}$ defined in (2.49) are related to $A$ and $C$ in (1.2) by

$$\begin{aligned} \widehat{A} &= T^{-1}AT, \\ \widehat{C} &= CT \end{aligned} \tag{2.53}$$

for some $T \in \mathbf{R}^{n\times n}$.

As noted before, in (2.48) $\widehat{\mathcal{H}}\mathcal{F}^\perp$ can be replaced with $R_{22}$.

**2.2. Extracting *B* and *D* from the data.** We will now determine $B$ and $D$ matrices in the realization using the given frequency domain data. Repeated application of the differentiation formula

$$\frac{d}{dz}X^{-1} = -X^{-1}\frac{dX}{dz}X^{-1}$$

to $X_{\mathrm{C}}(z) = (zI_n - A)^{-1}B$ yields the derivatives of $G(z)$ as follows:

$$\frac{d^j}{dz^j}G(z) = \delta_{0j}\, D + (-1)^j j!\, C(zI_n - A)^{-j-1}B, \qquad j \geq 0, \tag{2.54}$$

where $\delta_{ks}$ is the Kronecker delta. Now, let

$$\mathcal{G}_k \triangleq \begin{bmatrix} w_{k0} \\ w_{k1} \\ \vdots \\ w_{kN_k} \end{bmatrix}, \qquad k = 1, \ldots, L, \tag{2.55}$$

and

$$\mathcal{G} \triangleq \begin{bmatrix} \mathcal{G}_1 \\ \mathcal{G}_2 \\ \vdots \\ \mathcal{G}_L \end{bmatrix}. \tag{2.56}$$

Observe from (2.54) that, for fixed $A$ and $C$, the matrices $B$ and $D$ appear linearly in $\mathcal{G}$. Hence, we can uniquely determine $B$ and $D$ by solving the following linear least-squares problem

$$\widehat{B}, \widehat{D} = \arg\min_{B,D} \left\| \widehat{\mathcal{G}} - \widehat{\mathcal{Y}} \begin{bmatrix} B \\ D \end{bmatrix} \right\|_F^2, \tag{2.57}$$

where

$$\|X\|_F \triangleq \left[ \sum_k \sum_s |x_{ks}|^2 \right]^{1/2}$$

is the [illegible],

$$\widehat{\mathcal{G}} \triangleq \begin{bmatrix} \mathrm{Re}\mathcal{G} \\ \mathrm{Im}\mathcal{G} \end{bmatrix} \in \mathbf{R}^{pN \times m}, \tag{2.58}$$

$$\widehat{\mathcal{Y}} \triangleq \begin{bmatrix} \mathrm{Re}\mathcal{Y} \\ \mathrm{Im}\mathcal{Y} \end{bmatrix} \in \mathbf{R}^{pN \times (n+p)}, \tag{2.59}$$

and

$$\mathcal{Y}_k \triangleq \begin{bmatrix} C(z_k I_n - A)^{-1} & I_p \\ -C(z_k I_n - A)^{-2} & 0 \\ \vdots & \\ (-1)^{N_k} N_k!\, C(z_k I_n - A)^{-N_k - 1} & 0 \end{bmatrix}, \tag{2.60}$$

$$\mathcal{Y} \triangleq \begin{bmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \\ \vdots \\ \mathcal{Y}_L \end{bmatrix}, \tag{2.61}$$

provided that $\widehat{\mathcal{Y}}$ is not rank deficient. For the last requirement, a sufficient condition is presented next.

LEMMA 2.2. [illegible] $N$, [illegible] $\widehat{\mathcal{Y}}$ [illegible] (2.39), [illegible] (2.59) [illegible] $N > n$, [illegible] $A$ [illegible] $z_k$ [illegible]

$$\mathrm{rank}\widehat{\mathcal{Y}} = p + n \qquad \Longleftrightarrow \qquad (C, A) \text{ observable pair}. \tag{2.62}$$

The matrix $\mathcal{Y}$ is rank deficient if and only if there exists $\begin{bmatrix} B \\ D \end{bmatrix} \neq 0$ such that

$$\mathcal{Y}\begin{bmatrix} B \\ D \end{bmatrix} = 0 \iff \frac{d^j}{dz^j}G(z)\bigg|_{z=z_k} = 0, \quad 0 \leq j \leq N_k,\ k = 1, \dots, L.$$

As in the proof of Lemma 2.1, this equation implies that every element of $G(z)$ has a total number of $N$ zeros counting multiplicities, a contradiction if $G(z)$ is not identically zero unless $N \leq n$. ☐

Thus, from (2.53) and Lemma 2.2, if $N \geq q + n$ and $q > n$, we have

$$\widehat{B} = T^{-1}B, \qquad \widehat{D} = D. \tag{2.63}$$

Moreover,

$$\widehat{G}(z) \triangleq \widehat{C}(zI_n - \widehat{A})^{-1}\widehat{B} + \widehat{D} = G(z). \tag{2.64}$$

**2.3. Solvability conditions.** By picking $q = n + 1$ in the subspace-based algorithm developed above, we obtain a sufficient condition for the interpolation of $G(z)$ from its noise-free samples and derivatives evaluated at $L$ distinct points in $\vartheta$ as $N \geq 2n + 1$, where $N$ is defined by (2.39). This condition turns out to be a necessary condition for the interpolation of $G(z)$, as demonstrated next by a simple example.

Consider an $n$th-order stable single-input/single-output system represented by the transfer function

$$G(z) = \frac{b_0 z^n + b_1 z + \cdots + b_n}{z^n + a_1 z + \cdots + a_n}. \tag{2.65}$$

We are to determine $2n + 1$ unknown real coefficients $a_1, \dots, a_n, b_0, \dots, b_n$ from the evaluations of $G(z)$ and its derivatives at a given set of distinct frequencies $z_k \in \vartheta$. Let $N$ be as in (2.39).

Let us first assume in (1.4) that $N_k = 0$ and $z_k \in \mathbf{C} - \mathbf{R}$ for all $k$; i.e., the interpolation nodes are simple and purely complex numbers. Then, $N = 2L$. With $q = n + 1$, the subspace-based algorithm delivers a minimal realization of $G(z)$, provided that $2L \geq 2n + 1$. This condition is satisfied by choosing $L = n + 1$. Clearly, this is the least amount of data one could use to interpolate an arbitrary $n$th-order system, as can directly be verified by writing $2L$-linear equations down from (1.4) and (2.65) to determine the unknowns $a_1, \dots, a_n, b_0, \dots, b_n$. Notice that if some interpolation nodes have multiplicities, then the resulting equations become nonlinear in $a_1, \dots, a_n, b_0, \dots, b_n$.

Now, as a special case, let us consider the situation that all $z_k$ are on the unit circle excluding the points $\pm 1$. Thus, Algorithm 2.1 recovers $n$th-order stable systems from $n + 1$ noise-free frequency response measurements, excluding the frequencies 0 and $\pi$. If the frequencies contain 0, from (2.39) we then have $N = 2L - 1$. Hence, with $q = n + 1$ selected, we must have $2L - 1 \geq 2n + 1$, which is fulfilled by letting $L = n + 1$. If, in addition, the frequencies contain $\pi$ as well, we end up with the interpolation condition $L = n + 2$. The last conclusion extends an interpolation result in [24] derived for the uniformly spaced frequencies case to the nonuniformly spaced frequencies case.

It is easy to see, for example, by the partial fraction expansion or similar techniques, that these results hold for multi-input/multi-output systems with multiple interpolation nodes as well. Therefore, Algorithm 2.1 is capable of using a minimum amount of the frequency domain data for the Lagrange–Sylvester interpolation of stable systems.

**2.4. Summary of the subspace-based interpolation algorithm.** Let us summarize the interpolation algorithm in the following.

ALGORITHM 2.1.

1. (1.4) $\mathcal{H}$ $\mathcal{F}$ (2.35) (2.36) (2.30) (2.32) (2.24) (2.27) (2.22) (2.25) (2.15) (2.14)
2. (2.42)
3. (2.48) $\widehat{\mathcal{H}}\mathcal{F}^{\perp}$ $R_{22}$ (2.42)
4. $\widehat{\Sigma}_s$ $n$
5. $J_1$ $J_2$ $J_3$ (2.50)–(2.52) $\widehat{A}$ $\widehat{C}$ (2.49)
6. (2.57) $\widehat{B}$ $\widehat{D}$ $\widehat{\mathcal{G}}$ $\widehat{\mathcal{Y}}$ (2.58) (2.59) (2.60)–(2.61) (2.55)–(2.56)

Clearly, $\widehat{\Sigma}_o = 0$ in (2.48) when the data are not corrupted by noise, the system that has generated the data is of McMillan degree $n$, $N \geq q + n$, and $q > n$. As we stated earlier, Algorithm 2.1 produces a minimal stable realization of the interpolant, given that the latter exits. In most interpolation problems, the existence and the uniqueness questions are easily settled, and the construction of a solution (or all solutions) with certain properties such as the McMillan degree constraints, in particular minimality, remains a difficult one. The algorithm outlined above is straightforward to implement. In the implementation of the algorithm, it suffices to let $q = n + 1$ and $N = 2n + 1$, where $N$ is defined by (2.39). The system order, if unknown a priori, can be determined in step 4 of Algorithm 2.1 from the inspection of the singular values. This process also reveals redundancies in the data. Numerically, the most expensive step in the algorithm is the singular value decomposition of $R_{22}$. Notice with $q = n + 1$ and $N = 2n + 1$ selected, that $R_{22} \in \mathbf{R}^{p(n+1)\times mn}$.

The main result of this paper is captured in the following.

THEOREM 2.3. 2.1 (1.4) $n$ $N$ (2.39) $N \geq q + n$ $q > n$ $(\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$ $G(z)$

**2.5. Discussion.** In the rest of this section, we will briefly comment on the similarities and the differences between Algorithm 2.1 and the Löwner matrix–based approach [1].

The most striking difference between the methods appears to be the formation of data matrices. In [1], elements of a Löwner matrix are computed by taking partial derivatives of the divided differences $[G(z) - G(s)]/(z - s)$ evaluated at $z = z_k$ and $s = z_l$, where the number of the derivatives is determined by the particular choice of the (block) row and column sets and the multiplicities of the nodes. If $z_k$ equals $z_l$, a limiting process has to be used to define that particular element. It is required that the numbers of the chosen block rows and columns add up to $N$. The elements of $\mathcal{H}$ in the proposed algorithm, on the other hand, consist of linear combinations of the derivatives of the products $z^l G(z)$ evaluated at $z = z_k$, where for each $k$, $l$ satisfies $0 \leq l \leq N_k$. A simple transformation that relates $\mathcal{H}$ to a Löwner matrix does not seem possible unless all the $z_k$ are the same, in which case the problem solved reduces to a conventional partial realization problem. In the latter case, notice that this link is provided by the bilinear map (1.5).

Both algorithms rely on the factorization of the data matrices discussed above as a product of two matrices which are directly related to the observability and controllability concepts. In [1], the Löwner matrix is expressed as a product of the so-called generalized observability and the controllability matrices, whereas in the proposed algorithm this relation is recovered after some projections. In fact, the proofs of Lemmas 2.1, 2.2, and 3.1R in [1] use the same ideas.

The most striking similarity between the algorithms is the condition $N > 2n$. It should be noted that the stability assumption is not essential in the formulation of the interpolation problem, since the data are already assumed to originate from a finite-dimensional dynamical system with a complexity bounded above and the number of the nodes is finite. This assumption is necessary in an identification setup. Without the knowledge that the data have originated from a dynamical system with a complexity bounded above, the condition $N > 2n$ is precisely one of the requirements for the existence of a unique minimal-order interpolating rational matrix [1]. In addition to this requirement, there is also a more stringent rank condition captured in Assumption 4.1 in [1]. Thus, both algorithms operate under the same conditions which assure the existence of a unique minimal interpolating rational matrix. We have not addressed the properness issue in this paper due to our standing assumption on the origins of the data. Again, without the knowledge of the origins of the data, one has to secure that the solution of the interpolation problem is a proper transfer function. The properness is guaranteed by Assumption 4.2 in [1]. It is also noted there that this assumption can be eliminated by means of a suitably chosen bilinear transformation.

The Löwner matrix–based and proposed algorithms cannot be directly applied when there does not exist a unique minimal interpolating function and the data are not scalar. This may happen either in the presence of noise which corrupts transfer function evaluations or when the true dynamics is of higher dimension. The problem is then to find the admissible degrees of complexity, i.e., those positive integers $n$ for which there exist solutions $G(z)$ to the interpolation problem (1.4) with $deg\, G = n$, and to construct all corresponding solutions for a given admissible degree $n$. This problem is known as the partial realization problem. If the original data do not satisfy the criterion for the existence of a unique minimal interpolating function, one needs to add interpolation data until the criterion becomes satisfied. The fact that the data can be found so that the increase in degree is finite is nontrivial. The added data will necessarily drive up the degree of the interpolating transfer function. In the scalar case, dealt with in [2], the way this can be done is set out and is rather complicated. The multivariable case is studied in [4] using the generating system approach. While [4] gives the theory behind the determination of the minimal McMillan degree and all admissible degrees, the current paper and [1] provide the theory behind the construction in state-space terms of the solution of admissible degrees.

A departure of Algorithm 2.1 from the Löwner matrix–based approach is the determination of the minimal order. Under the stated conditions, in Algorithm 2.1 the minimal order and the observability range space are extracted by a singular value decomposition, while in the Löwner matrix–based approach the minimal order is determined by checking ranks of several (generalized) Löwner matrices. The singular value decomposition is not sensitive to random inaccuracies in data; that is, the true singular values and the observability range space are consistently estimated as $N$ increases unboundedly, provided that $n$ is finite or increases more slowly than $N$ [24, 25]. To our best knowledge, an asymptotic error analysis for randomly corrupted transfer function evaluations has not been performed for any of the interpolation algorithms in the literature.

Deficiencies of the proposed interpolation algorithm and the Löwner matrix–based approach are the same. As pointed out in [1], a parameterization of solutions when the original data have to be added and derivation of recursive formulae for allowing update of a realization when one or more interpolation data become available are absent. It would be interesting to develop connections between the constrained interpolation problems such as the Nevanlinna–Pick and the positive-real interpolation and Algorithm 2.1. It is worth mentioning that the Nevanlinna–Pick interpolation can be transformed into an interpolation problem without norm constraint by adding the mirror image interpolation points to the original data [3].

**3. Subspace-based identification with interpolation constraints.** In this section, we will consider identification of an $n$th-order stable system with transfer function $G(z)$ from noisy samples of the frequency response,

$$w_l = G(e^{i\theta_l}) + \eta_l, \qquad l = 1, \dots, M, \tag{3.1}$$

with the interpolation constraints

$$\left.\frac{d^j}{dz^j} G(z)\right|_{z=z_k} = E_{kj}, \qquad j = 0, 1, \dots, N_k, \quad k = 1, \dots, L, \tag{3.2}$$

where $0 \leq \theta_l \leq \pi$, $l = 1, \dots, M$, denote the discrete-time frequencies and $\eta_l$ is a sequence of independent zero-mean complex random variables with a known covariance function that is uniformly bounded. The number of the constraints defined in (2.39) satisfies $N < n$. The interpolation constraints (3.2) reflect the prior knowledge on $G(z)$. For example, by taking $E_{kj} = 0$ for all $j \leq N_k$, we enforce a zero with multiplicity $N_k + 1$ at $z_k$. These constraints may also be used as design variables to focus on a frequency band of interest.

We would like to find an identification algorithm which maps the data $\{w_l, \theta_l\}_{l=1}^{M}$ to an $n$th-order model $\widehat{G}_M(z)$ that satisfies the interpolation constraints in (3.2) such that, with probability one,

$$\lim_{M\to\infty} \|\widehat{G}_M - G\|_\infty = 0,$$

where

$$\|X\|_\infty \triangleq \sup_\omega \sigma_1(X(e^{i\omega}))$$

and $\sigma_1$ denotes the largest singular value. Algorithms with this property are called [illegible]. This identification setup except for the constraints in (3.2) can be found, for example, in [24].

A motivating example for the constraints in (3.2) is as follows. Suppose that the system to be identified is $n$th order stable single-input/single-output continuous-time system represented by the transfer function

$$G^c(s) = \frac{b_0 s^m + b_1 s + \cdots + b_m}{s^n + a_1 s + \cdots + a_n}, \tag{3.3}$$

where the denominator degree $n$ is greater than the numerator degree $m$, and we are given $M$ noise corrupted frequency response measurements

$$w_l = G(i\omega_l) + \eta_l, \qquad l = 1, \dots, M. \tag{3.4}$$

Assuming $b_0 \neq 0$, the [illegible] of $G^c(s)$ is defined as $\tau \triangleq n - m$.

A direct use of the Möbius transform technique (1.5) targets identifying the discrete-time equivalent of $G^c(s)$ defined by

$$G^d(z) \triangleq G^c(\psi(z)), \tag{3.5}$$

using $w_l$, $l = 1, \ldots, M$, at the transformed discrete-time frequencies

$$\theta_k = 2 \arctan\left(\frac{\omega_k}{\lambda}\right), \qquad k = 1, \ldots, M. \tag{3.6}$$

Then, the continuous-time identified transfer function denoted by $\widehat{G}^c_M(s)$ is obtained from the discrete-time identified transfer function denoted by $\widehat{G}^d_M(z)$ by using the inverse Möbius map $z = \psi^{-1}(s)$; i.e., $\widehat{G}^c_M(s) = \widehat{G}^d_M(\psi^{-1}(s))$. Due to noise and unmodeled dynamics, the former is only a proper transfer function.

If maintaining the relative degree is a concern, we then high-pass filter $\widehat{G}^c_M(s)$ as follows:

$$\widehat{G}_M(s) = \frac{\widehat{G}^c_M(s)}{(s+\mu)^\tau},$$

where $\mu > 0$ is chosen sufficiently outside the bandwidth of $\widehat{G}^c_M(s)$. This filtering increases the order of the identified model by $\tau$. This problem can be circumvented by including the constraints

$$\left.\frac{d^j}{dz^j} G^d(z)\right|_{z=-1} = 0, \qquad j = 1, \ldots, \tau,$$

in the problem formulation. Observe that when applied to (3.3), the Möbius map (1.5) introduces a zero of $G^d(z)$ at $z = -1$ with multiplicity $\tau$.

Now, the solution of the constrained identification problem (3.1)–(3.2) is particularly simple if one notes from (2.54) the following set of equations:

$$\delta_{0j}\, D + (-1)^j j!\, C(z_k I_n - A)^{-j-1} B = E_{kj}, \;\; j = 0, \ldots, N_k, \;\; k = 1, \ldots, L, \tag{3.7}$$

which describe $N$ hyperplanes in the parameter space of $B$ and $D$ for fixed $C$ and $A$. Hence, it suffices to solve the linear least-squares problem (2.57) with the linear constraints (3.7). With this modification, the frequency domain subspace-based identification algorithm presented in [24] is strongly consistent. The inclusion of the noise covariance information in the algorithm is straightforward and can be found in [24]. This extension can be viewed as the tangential version of the Lagrange–Sylvester interpolation problem (1.1).

**4. Example.** The purpose of this section is to illustrate Algorithm 2.1 with a step-by-step numerical example. Suppose that the system to be found by interpolation has the following state-space representation:

$$A = \begin{bmatrix} -0.5 & 0.5 & 0 & 0 \\ -0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & -0.25 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Thus, $n = 4$, $p = 2$, and $m = 3$. This system has the transfer function

$$G(z) = \begin{bmatrix} \dfrac{z^2 + 3z + 1.5}{z^2 + z + 0.5} & -\dfrac{z^3 + 0.5z^2 + 0.5z + 0.75}{z^3 + 0.5z^2 - 0.25} & 0 \\ \dfrac{2z^2 + 1.25z + 0.5}{z^3 + 1.25z^2 + 0.75z + 0.125} & \dfrac{z^3 + 3.25z^2 + 2.5z + 0.75}{z^3 + 1.25z^2 + 0.75z + 0.125} & \dfrac{z + 1.25}{z + 0.25} \end{bmatrix}.$$

Let us assume that the interpolation data are as follows:

$$z_1 = 1 + i,\ N_1 = 0,\ z_2 = 1 - i,\ N_2 = 0,\ z_3 = 2,\ N_3 = 4$$

and

$$w_{10} = \begin{bmatrix} 1.9333 - 0.5333i & -0.8667 + 0.4000i & 0 \\ 0.8878 - 0.5236i & 1.9545 - 0.6569i & 1.4878 - 0.3902i \end{bmatrix},$$

$$w_{20} = \begin{bmatrix} 1.9333 + 0.5333i & -0.8667 - 0.4000i & 0 \\ 0.8878 + 0.5236i & 1.9545 + 0.6569i & 1.4878 + 0.3902i \end{bmatrix},$$

$$w_{30} = \begin{bmatrix} 1.7692 & -1.2051 & 0 \\ 0.7521 & 1.8291 & 1.4444 \end{bmatrix}, \quad w_{31} = \begin{bmatrix} -0.2840 & 0.2433 & 0 \\ -0.2804 & -0.3395 & -0.1975 \end{bmatrix},$$

$$w_{32} = \begin{bmatrix} 0.2003 & -0.4251 & 0 \\ 0.2084 & 0.2757 & 0.1756 \end{bmatrix}, \quad w_{33} = \begin{bmatrix} -0.2000 & 0.9844 & 0 \\ -0.2333 & -0.3341 & -0.2341 \end{bmatrix},$$

$$w_{34} = \begin{bmatrix} 0.2456 & -2.8518 & 0 \\ 0.3531 & 0.5390 & 0.4162 \end{bmatrix}.$$

Then we set $q = 5$ and compute $N = 9$. Therefore, the inequalities $N \geq q + n$ and $q > n$ are both satisfied. In step 1, we compute the matrices $\widehat{\mathcal{H}} \in \mathbf{R}^{10\times 27}$ and $\mathcal{F} \in \mathbf{R}^{15\times 27}$. The QR-factorization in step 2 results in $R_{22} \in \mathbf{R}^{10\times 12}$ given by

$$R_{22} = \left[\begin{array}{rrrrccc} -0.4622 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0.0381 & -0.0518 & 0 & 0 & \vdots & \ddots & \vdots \\ -0.2544 & 0.0203 & -0.0240 & 0 & & & \\ -0.0176 & 0.0194 & -0.0075 & -0.0009 & & & \\ -0.1144 & -0.0070 & 0.0091 & -0.0089 & & & \\ 0.0094 & -0.0110 & 0.0094 & -0.0025 & & & \\ -0.0583 & 0.0035 & -0.0045 & 0.0045 & & & \\ -0.0033 & 0.0057 & -0.0061 & 0.0037 & & & \\ -0.0344 & 0.0033 & -0.0037 & -0.0022 & \vdots & \ddots & \vdots \\ -0.0007 & -0.0013 & 0.0015 & -0.0027 & 0 & \cdots & 0 \end{array}\right],$$

which is not unexpected since $n = 4$. In step 3, we compute the nonzero singular values 0.5460, 0.0609, 0.0249, and 0.0098. The matrices $\widehat{A}$ and $\widehat{C}$ computed in step 5 are

$$\widehat{A} = \begin{bmatrix} 0.5204 & -0.1361 & 0.3199 & 0.5352 \\ 0.0882 & -0.4983 & 0.4848 & -0.1035 \\ 0.0052 & 0.0820 & -0.4810 & 0.7195 \\ -0.0295 & 0.1919 & -0.3546 & -0.2911 \end{bmatrix},$$

$$\widehat{C} = \begin{bmatrix} 0.8460 & 0.2123 & -0.2149 & -0.3233 \\ -0.0721 & 0.8069 & 0.5289 & 0.1046 \end{bmatrix}.$$

In step 6, we compute $\widehat{\mathcal{G}} \in \mathbf{R}^{18\times 3}$ and $\widehat{\mathcal{Y}} \in \mathbf{R}^{18\times 6}$ matrices, and the solution of the least-squares problem is

$$\widehat{B} = \begin{bmatrix} 1.0502 & -0.5390 & -0.0816 \\ 2.8626 & 1.8321 & 0.9041 \\ -0.1545 & 1.0984 & 0.4896 \\ -1.4555 & -0.9375 & 0.0547 \end{bmatrix},$$

$$\widehat{D} = \begin{bmatrix} 1.0000 & -1.0000 & -0.0000 \\ -0.0000 & 1.0000 & 1.0000 \end{bmatrix}.$$

The realization $(\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D})$ is similar to $(A, B, C, D)$. In fact, the estimates of the interpolation data computed from the former has a maximum error $5.9746 \times 10^{-14}$.

**5. Conclusions.** In this paper, we presented a new algorithm for the Lagrange–Sylvester interpolation of rational matrix functions that are analytic at infinity. This algorithm is related to the recent frequency domain subspace-based identification methods and is not sensitive to inaccuracies in data. A necessary and sufficient condition for the existence and the uniqueness of a minimal interpolant was formulated in terms of the total multiplicity of the interpolation nodes. The purpose of this contribution was to pinpoint the kinship between the frequency domain subspace-based identification of stable linear systems and the minimal rational interpolation of stable systems.

**Acknowledgment.** The authors would like to thank the associate editor and the reviewers of the manuscript for helpful remarks and references.

## REFERENCES

[1] B. D. O. Anderson and A. C Antoulas, *Rational interpolation and state-variable realizations*, Linear Algebra Appl., 137/138 (1990), pp. 479–509.

[2] A. C. Antoulas and B. D. O. Anderson, *On the scalar rational interpolation problem*, IMA J. Math. Control Inform., 3 (1986), pp. 61–88.

[3] A. C. Antoulas and B. D. O. Anderson, *On the problem of stable rational interpolation*, Linear Algebra Appl., 122/123/124 (1989), pp. 301–329.

[4] A. C Antoulas, J. A. Ball, J. Kang, and J. C. Willems, *On the solution of the minimal rational interpolation problem*, Linear Algebra Appl., 137/138 (1990), pp. 511–573.

[5] J. A. Ball and J. Kang, *Matrix polynomial solution of tangential Lagrange-Sylvester interpolation conditions of low McMillan degree*, Linear Algebra Appl., 137/138 (1990), pp. 699–746.

[6] J. A. Ball, I. Gohberg, and L. Rodman, *Interpolation of Rational Matrix Functions*, Birkhäuser, Basel, Switzerland, 1990.

[7] J. A. Ball, I. Gohberg, and L. Rodman, *Two-sided Lagrange-Sylvester interpolation problem for rational matrix functions*, in Proc. Sympos. Pure Math. 51, AMS, Providence, RI, 1990, pp. 17–83.

[8] J. A. Ball, I. Gohberg, and L. Rodman, *Simultaneous residue interpolation for rational matrix functions*, Integral Equations Operator Theory, 13 (1990), pp. 611–637.

[9] J. A. Ball, M. A. Kaashoek, G. Groenewald, and J. Kim, *Column reduced rational matrix functions with given null-pole data in the complex plane*, Linear Algebra Appl., 203/204 (1994), pp. 111–138.

[10] A. Blomqvist, A. Lindquist, and R. Nagamune, *Matrix-valued Nevanlinna-Pick interpolation with complexity constraints: An optimization approach*, IEEE Trans. Automat. Control, 48 (2003), pp. 2172–2190.

[11] T. Boros, A. H. Sayed, and T. Kailath, *A recursive method for solving unconstrained tangential interpolation problems*, IEEE Trans. Automat. Control, 44 (1999), pp. 454–470.

[12] C. I. Byrnes, T. T. Georgiou, and A. Lindquist, *A new approach to spectral estimation: A tunable high-resolution spectral estimator*, IEEE Trans. Signal Process., 48 (2000), pp. 3189–3205.

[13] J. Chen and G. Gu, *Control Oriented System Identification: An $H_\infty$ Approach*, Wiley-Interscience, New York, 2000.

[14] I. P. Fedcina, *A description of the solution of the Nevanlinna-Pick tangent problem*, Akad. Nauk Armyan. SSR. Dokl., 60 (1975), pp. 37–42 (in Russian).

[15] I. P. Fedcina, *The Nevanlinna-Pick tangent problem with multiple points*, Akad. Nauk Armyan. SSR. Dokl., 61 (1975), pp. 214–218 (in Russian).

[16] T. Kailath, *Linear Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1980.

[17] R. E. Kalman, *On partial realization, transfer-functions and canonical forms*, Acta. Polytech. Scand. Math., 31 (1979), pp. 9–32.

[18] J. Kamali, T. Boros, T. Kailath, and G. Franklin, *Q-parametrization for unstable plants: A displacement structure approach*, in Proceedings of the American Control Conference, 1995, pp. 4401–4402.

[19] H. Kimura, *Directional interpolation approach to $H^\infty$-optimization and robust stabilization*, IEEE Trans. Automat. Control, AC-32 (1987), pp. 1085–1093.

[20] H. Langer and A. Lasarow, *Solution of a multiple Nevanlinna-Pick problem via orthogonal rational functions*, J. Math. Anal. Appl., 293 (2004), pp. 605–632.

[21] D. J. N. Limebeer and B. D. O. Anderson, *An interpolation theory approach to $H^\infty$-controller degree bounds*, Linear Algebra Appl., 98 (1988), pp. 347–386.

[22] L. Ljung, *Linear system identification as curve fitting*, in Directions in Mathematical Systems Theory and Optimization, A. Rantzer and C. I. Byrnes, eds., LNCIS 286, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 203–215.

[23] K. Liu, R. N. Jacgues, and D. W. Miller, *Frequency domain structural system identification by observability range space extraction*, in Proceedings of the American Control Conference, Baltimore, MD, 1994, pp. 107–111.

[24] T. McKelvey, H. Akçay, and L. Ljung, *Subspace-based multivariable system identification from frequency response data*, IEEE Trans. Automat. Control, 41 (1996), pp. 960–979.

[25] T. McKelvey, H. Akçay, and L. Ljung, *Subspace-based identification of infinite-dimensional multivariable systems from from frequency response data*, Automatica, 32 (1996), pp. 885–902.

[26] C. Mosquera and F. Pérez, *Algebraic solution to the robust SPR problem for two polynomials*, Automatica, 37 (2001), pp. 757–762.

[27] Nikolai K. Nikolski, *Operators, Functions, and Systems: An Easy Reading, Vol. 1: Hardy, Hankel, and Toeplitz*, AMS, Providence, RI, 2002.

[28] Nikolai K. Nikolski, *Operators, Functions, and Systems: An Easy Reading, Vol. 2: Model Operators and Systems*, AMS, Providence, RI, 2002.

[29] K. Y. Osipenko, *Optimal Recovery of Analytic Functions*, Nova Science Publishers, New York, 2000.

[30] A. H. Sayed, T. Kailath, H. Lev-Ari, and T. Constantinescu, *Recursive solutions of rational interpolation problems via fast matrix factorization*, Integral Equations Operator Theory, 20 (1994), pp. 84–118.

[31] A. H. Sayed, T. Constantinescu, and T. Kailath, *Time-variant displacement structure and interpolation problems*, IEEE Trans. Automat. Control, 39 (1994), pp. 960–976.

[32] R. S. Smith and J. C. Doyle, *Model validation: A connection between robust control and identification*, IEEE Trans. Automat. Control, 37 (1992), pp. 942–952.

[33] P. Van Overschee and B. De Moor, *Continuous-time frequency domain subspace identification*, Signal Processing, 52 (1996), pp. 179–194.

[34] D. C. Youla and M. Saito, *Interpolation with positive-real functions*, J. Franklin Inst., 284 (1967), pp. 77–108.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 396–412

# ITERATIVE SOLUTION OF A NONSYMMETRIC ALGEBRAIC RICCATI EQUATION*

CHUN-HUA GUO† AND NICHOLAS J. HIGHAM‡

**Abstract.** We study the nonsymmetric algebraic Riccati equation whose four coefficient matrices are the blocks of a nonsingular $M$-matrix or an irreducible singular $M$-matrix $M$. The solution of practical interest is the minimal nonnegative solution. We show that Newton's method with zero initial guess can be used to find this solution without any further assumptions. We also present a qualitative perturbation analysis for the minimal solution, which is instructive in designing algorithms for finding more accurate approximations. For the most practically important case, in which $M$ is an irreducible singular $M$-matrix with zero row sums, the minimal solution is either stochastic or substochastic and the Riccati equation can be transformed into a unilateral matrix equation by a procedure of Ramaswami. The minimal solution of the Riccati equation can then be found by computing the minimal nonnegative solution of the unilateral equation using the Latouche–Ramaswami algorithm. When the minimal solution of the Riccati equation is stochastic, we show that the Latouche–Ramaswami algorithm, combined with a shift technique suggested by He, Meini, and Rhee, is breakdown-free and is able to find the minimal solution more efficiently and more accurately than the algorithm without a shift. When the minimal solution of the Riccati equation is substochastic, we show how the substochastic minimal solution can be found by computing the stochastic minimal solution of a related Riccati equation of the same type.

**Key words.** nonsymmetric algebraic Riccati equation, $M$-matrix, minimal nonnegative solution, perturbation analysis, Newton's method, Latouche–Ramaswami algorithm, shifts

**AMS subject classifications.** 15A24, 15A48, 65F30, 65H10

**DOI.** 10.1137/050647669

## 1. Introduction.

We consider the nonsymmetric algebraic Riccati equation (or NARE)

$$\mathcal{R}(X) = XCX - XD - AX + B = 0, \tag{1.1}$$

where $A, B, C, D$ are real matrices of sizes $m \times m, m \times n, n \times m, n \times n$, respectively, and we assume throughout that

$$M = \begin{bmatrix} D & -C \\ -B & A \end{bmatrix} \tag{1.2}$$

is a nonsingular $M$-matrix or an irreducible singular $M$-matrix. Some relevant definitions are as follows. For any matrices $A, B \in \mathbb{R}^{m\times n}$, we write $A \geq B$ ($A > B$) if $a_{ij} \geq b_{ij} (a_{ij} > b_{ij})$ for all $i, j$. A real square matrix $A$ is called a $Z$-matrix if all its off-diagonal elements are nonpositive. It is clear that any $Z$-matrix $A$ can be written as $sI - B$ with $B \geq 0$. A $Z$-matrix $A$ is called an $M$-matrix if $s \geq \rho(B)$, where $\rho(\cdot)$ is the spectral radius; it is a singular $M$-matrix if $s = \rho(B)$ and a nonsingular $M$-matrix if $s > \rho(B)$.

---

[*] Received by the editors December 15, 2005; accepted for publication (in revised form) by Q. Ye October 3, 2006; published electronically March 22, 2007. This work was supported in part by a Royal Society-Wolfson Research Merit Award to the second author.

http://www.siam.org/journals/simax/29-2/64766.html

[†] Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca, http://www.math.uregina.ca/∼chguo/). The work of this author was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

[‡] School of Mathematics, The University of Manchester, Sackville Street, Manchester, M60 1QD, UK (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/∼higham/).

The NARE (1.1) has applications in transport theory and Markov models [20, 27, 28]. The solution of practical interest is the minimal nonnegative solution. The equation has attracted much attention recently [1, 4, 10, 11, 14, 16, 17, 18, 21, 24, 26].

For application to Markov models, the case of primary interest is the one where $M$ is an irreducible singular $M$-matrix with zero row sums. When $M$ is an irreducible singular $M$-matrix, we have $M = \rho(N)I - N$ for some irreducible nonnegative matrix $N$. Thus, by applying the Perron–Frobenius theorem to $N$, there are positive vectors $u_1, v_1 \in \mathbb{R}^n$ and $u_2, v_2 \in \mathbb{R}^m$ such that

$$M(v_1^T \;\, v_2^T)^T = 0, \quad (u_1^T \;\, u_2^T)M = 0, \tag{1.3}$$

and the vectors $(v_1^T \;\, v_2^T)$ and $(u_1^T \;\, u_2^T)$ are each unique up to a scalar multiple.

Since $M$ is a nonsingular $M$-matrix or an irreducible singular $M$-matrix, we have $B, C \geq 0$, and $A$ and $D$ are nonsingular $M$-matrices (see [10], for example). Therefore, the matrix $I \otimes A + D^T \otimes I$ is also a nonsingular $M$-matrix, where $\otimes$ is the Kronecker product. Some properties of the NARE (1.1) are summarized below. See [10, 11, 13] for more details.

THEOREM 1.1. [illegible] $M$ [illegible] $M$[illegible] [illegible] [illegible] $M$[illegible] [illegible] (1.1) [illegible] $S$[illegible] $M$ [illegible] $S > 0$ [illegible] $A - SC$ [illegible] $D - CS$ [illegible] $M$[illegible] [illegible] $M$ [illegible] $M$[illegible] [illegible] $A - SC$ [illegible] $D - CS$ [illegible] $M$[illegible] [illegible] $M$ [illegible] $M$[illegible] [illegible] $M$[illegible] [illegible] $u_1^T v_1 \neq u_2^T v_2$ [illegible]

$$M_S = I \otimes (A - SC) + (D - CS)^T \otimes I$$

[illegible] $M$[illegible] [illegible] $M$ [illegible] $M$[illegible] [illegible] $u_1^T v_1 = u_2^T v_2$ [illegible] $M_S$ [illegible] $M$[illegible]

We will also need the dual equation of (1.1):

$$YBY - YA - DY + C = 0. \tag{1.4}$$

This equation has the same type as (1.1): the matrix

$$\begin{bmatrix} A & -B \\ -C & D \end{bmatrix}$$

is a nonsingular $M$-matrix or an irreducible singular $M$-matrix if and only if the matrix $M$ has the same property. The minimal nonnegative solution of (1.4) is denoted by $\widehat{S}$.

A number of numerical methods have been studied for finding the minimal solution $S$, some of which require additional assumptions on the NARE (1.1). In particular, a class of basic fixed-point iterations has been studied in [10] and [16]. The Schur method has been studied in [10] and a modified Schur method is given in [14]. These methods are applicable without further assumptions on (1.1). Newton's method has also been studied in [10] and [16], where convergence of the Newton sequence $\{X_k\}$, with $X_0 = 0$, to the minimal solution $S$ has been established under the additional assumption that

$$B, C \neq 0, \quad (I \otimes A + D^T \otimes I)^{-1}\text{vec}B > 0. \tag{1.5}$$

Here, the vec operator stacks the columns of a matrix into one long vector. When $M$ is irreducible, we have $B, C \neq 0$. However, the condition $(I \otimes A + D^T \otimes I)^{-1}\text{vec}B > 0$

is not guaranteed by the irreducibility of $M$, as is shown in [10]. The question then arises as to whether (1.5) is necessary for the convergence of the Newton iteration. Our first contribution in this paper is a proof of convergence without this additional condition.

When $M$ is an irreducible singular $M$-matrix and $u_1^T v_1 = u_2^T v_2$, the matrix $M_S$ is a singular $M$-matrix. In this case, Newton's method has a singular Jacobian at the solution, and thus we cannot expect to find an accurate solution by the Newton iteration in finite precision arithmetic. A modified Schur method has been proposed in [14] to find a more accurate solution when $u_1^T v_1 \approx u_2^T v_2$. Another approach is to transform the bilateral equation (1.1) into a unilateral equation and use methods based on cyclic reduction, including the Latouche–Ramaswami (LR) algorithm [23], in combination with a shift technique proposed in [19].

The design of numerical methods for finding the minimal solution with higher accuracy is related to the perturbation behavior of the minimal solution. The minimal solution $S$ is a function of $M$ in (1.2). If the matrix $M$ is perturbed to $\widetilde{M}$, which is always assumed to be again a nonsingular $M$-matrix or an irreducible singular $M$-matrix, and $\widetilde{S}$ is the new minimal solution, we would like to know the relation between $\|\widetilde{S} - S\|$ and $\|\widetilde{M} - M\|$, where $\|\cdot\|$ is any matrix norm. Our second contribution is to prove the following.

- If $M$ is a nonsingular $M$-matrix or an irreducible singular $M$-matrix with $u_1^T v_1 \neq u_2^T v_2$, then there exist constants $\gamma > 0$ and $\epsilon > 0$ such that $\|\widetilde{S} - S\| \leq \gamma \|\widetilde{M} - M\|$ for all $\widetilde{M}$ with $\|\widetilde{M} - M\| < \epsilon$.
- If $M$ is an irreducible singular $M$-matrix with $u_1^T v_1 = u_2^T v_2$, then there exist constants $\gamma > 0$ and $\epsilon > 0$ such that
  - (a) $\|\widetilde{S} - S\| \leq \gamma \|\widetilde{M} - M\|^{1/2}$ for all $\widetilde{M}$ with $\|\widetilde{M} - M\| < \epsilon$;
  - (b) $\|\widetilde{S} - S\| \leq \gamma \|\widetilde{M} - M\|$ for all singular $\widetilde{M}$ with $\|\widetilde{M} - M\| < \epsilon$.

This result tells us that to achieve high accuracy for $S$ when $M$ is an irreducible singular $M$-matrix with $u_1^T v_1 \approx u_2^T v_2$, it is necessary to use the singularity of $M$ in the design of algorithms. Otherwise, we can only expect to achieve an accuracy of $O(\epsilon_m^{1/2})$, where $\epsilon_m$ is the machine epsilon. The modified Schur method in [14] and the methods using a shift technique in [4] and [14] all use the singularity of $M$. However, the use of the shift technique creates a new problem: it is not clear whether the resulting algorithm may break down, although quadratic convergence is guaranteed if no breakdown occurs. Our third contribution is to show that the (simplified) LR algorithm with a shift technique, presented in [14], is breakdown-free.

## 2. Convergence of Newton's method.

The Riccati function $\mathcal{R}$ is a mapping from $\mathbb{R}^{m\times n}$ into itself. The Fréchet derivative of $\mathcal{R}$ at a matrix $X$ is a linear map $\mathcal{R}'_X : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ given by

$$\mathcal{R}'_X(Z) = -\big((A - XC)Z + Z(D - CX)\big). \tag{2.1}$$

The Newton method for the solution of (1.1) is

$$X_{i+1} = X_i - (\mathcal{R}'_{X_i})^{-1}\mathcal{R}(X_i), \quad i = 0, 1, \ldots, \tag{2.2}$$

where the maps $\mathcal{R}'_{X_i}$ all need to be nonsingular. In view of (2.1), the iteration (2.2) is equivalent to

$$(A - X_i C)X_{i+1} + X_{i+1}(D - CX_i) = B - X_i C X_i, \quad i = 0, 1, \ldots . \tag{2.3}$$

We will need the following well-known result (see [2], for example).

THEOREM 2.1. [illegible] $Z$ [illegible] $A$ [illegible]

(a) $A$ [illegible] $M$ [illegible]

(b) $A^{-1} \geq 0$

(c) $Av > 0$ [illegible] $v > 0$

(d) [illegible] $A$ [illegible]

The equivalence of (a) and (c) in Theorem 2.1 implies the next result.

LEMMA 2.2. [illegible] $A$ [illegible] $M$ [illegible] $B \geq A$ [illegible] $Z$ [illegible] $B$ [illegible] $M$ [illegible]

We can now give a proof of convergence of the Newton iteration that does not require the assumption (1.5) made in [10].

THEOREM 2.3. [illegible] $S$ [illegible] (1.1) [illegible] (2.3) [illegible] $X_0 = 0$ [illegible] $\{X_i\}$ [illegible] $X_k \leq X_{k+1} \leq S$ [illegible] $k \geq 0$ [illegible] $\lim_{i\to\infty} X_i = S$

[illegible] Throughout the proof, we use the notation

$$M_X = I \otimes (A - XC) + (D - CX)^T \otimes I$$

for a given matrix $X$ (this notation is consistent with the notation $M_S$ already used in Theorem 1.1). Since $S$ is a solution of (1.1),

$$SCS - SD - AS + B = 0. \tag{2.4}$$

For the Newton iteration (2.3) with $X_0 = 0$, we have $AX_1 + X_1D = B$, which is equivalent to

$$(I \otimes A + D^T \otimes I)\mathrm{vec}X_1 = \mathrm{vec}B. \tag{2.5}$$

Since $I \otimes A + D^T \otimes I$ is a nonsingular $M$-matrix, Theorem 2.1(b) and (2.5) imply $\mathrm{vec}X_1 \geq 0$, i.e., $X_1 \geq 0$.

We first assume that $M$ is a nonsingular $M$-matrix, and we will prove by induction that

$$X_k \leq X_{k+1}, \quad X_k \leq S, \quad M_{X_k} \text{ is a nonsingular } M\text{-matrix} \tag{2.6}$$

for $k \geq 0$. It is clear that (2.6) is true for $k = 0$. We now assume that (2.6) is true for $k = i \geq 0$. By (2.3) and (2.4) we have

$$\begin{aligned}(A - X_iC)(X_{i+1} - S) + (X_{i+1} - S)(D - CX_i) \\ &= B - X_iCX_i - AS + X_iCS - SD + SCX_i \\ &= -(S - X_i)C(S - X_i).\end{aligned} \tag{2.7}$$

Since $X_i \leq S$ and $M_{X_i}$ is a nonsingular $M$-matrix, it follows from Theorem 2.1(b) and (2.7) that $X_{i+1} \leq S$. Since $M_S$ is a nonsingular $M$-matrix by Theorem 1.1, it follows from Lemma 2.2 that $M_{X_{i+1}}$ is a nonsingular $M$-matrix. By (2.3),

$$\begin{aligned}&(A - X_{i+1}C)X_{i+1} + X_{i+1}(D - CX_{i+1}) \\ &= \big(A - X_iC - (X_{i+1} - X_i)C\big)X_{i+1} + X_{i+1}\big(D - CX_i - C(X_{i+1} - X_i)\big) \\ &= B - X_iCX_i - (X_{i+1} - X_i)CX_{i+1} - (X_i + X_{i+1} - X_i)C(X_{i+1} - X_i) \\ &= B - X_{i+1}CX_{i+1} - (X_{i+1} - X_i)C(X_{i+1} - X_i).\end{aligned} \tag{2.8}$$

By (2.8) and (2.3),

$$\begin{aligned}&(A - X_{i+1}C)(X_{i+1} - X_{i+2}) + (X_{i+1} - X_{i+2})(D - CX_{i+1})\\ &\qquad = -(X_{i+1} - X_i)C(X_{i+1} - X_i) \le 0.\end{aligned}$$

Therefore, $X_{i+1} \le X_{i+2}$. We have thus proved that (2.6) is true for $k = i + 1$. Hence (2.6) is true for all $k \ge 0$ by induction.

We now assume that $M$ is an irreducible singular $M$-matrix. Then $S > 0$ by Theorem 1.1. Thus, the statement

$$X_k \le X_{k+1}, \quad X_k < S, \quad M_{X_k} \text{ is a nonsingular } M\text{-matrix} \tag{2.9}$$

is true for $k = 0$. Assume that (2.9) is true for $k = i \ge 0$. Then, by (2.7) we get $X_{i+1} < S$. It follows from (2.8) and (2.4) that

$$\begin{aligned}&(A - X_{i+1}C)(X_{i+1} - S) + (X_{i+1} - S)(D - CX_{i+1})\\ &\qquad = -(X_{i+1} - X_i)C(X_{i+1} - X_i) - (X_{i+1} - S)C(X_{i+1} - S) < 0.\end{aligned}$$

Therefore, $M_{X_{i+1}}\mathrm{vec}(S - X_{i+1}) > 0$. Thus $M_{X_{i+1}}$ is a nonsingular $M$-matrix by Theorem 2.1(c). It follows as before that $X_{i+1} \le X_{i+2}$. So (2.9) is true for $k = i + 1$, and hence for all $k \ge 0$ by induction.

Therefore, in both cases, the Newton sequence $X_k$ is well defined, monotonically increasing, and bounded above by $S$. Let $\lim_{k\to\infty} X_k = X_*$. Then $X_*$ is a nonnegative solution of (1.1) by (2.3). Since $X_* \le S$ and $S$ is minimal, we have $X_* = S$. ☐

**3. Perturbation analysis for the minimal solution.** In this section we are interested in a qualitative description of the perturbation of the minimal nonnegative solution $S$ of (1.1) as a function of $M$. The perturbation analysis of the minimal solution will be carried out through the perturbation analysis of a proper invariant subspace of the matrix

$$L = \begin{bmatrix} D & -C \\ B & -A \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} M. \tag{3.1}$$

Let all eigenvalues of $L$ be arranged in descending order of their real parts and be denoted by $\lambda_1, \ldots, \lambda_n, \lambda_{n+1}, \ldots, \lambda_{n+m}$. Then (see [10])

$$\sigma(D - CS) = \{\lambda_1, \ldots, \lambda_n\}$$

and

$$\sigma(A - SC) = \sigma(A - B\widehat{S}) = \{-\lambda_{n+1}, \ldots, -\lambda_{n+m}\}, \tag{3.2}$$

where $\widehat{S}$ is the minimal nonnegative solution of the dual equation (1.4). If $M$ is a nonsingular $M$-matrix, then $\lambda_1, \ldots, \lambda_n \in \mathbb{C}^+$ (the open right half plane) and $\lambda_{n+1}, \ldots, \lambda_{n+m} \in \mathbb{C}^-$ (the open left half plane). If $M$ is an irreducible singular $M$-matrix, then $\lambda_1, \ldots, \lambda_{n-1} \in \mathbb{C}^+$, $\lambda_{n+2}, \ldots, \lambda_{n+m} \in \mathbb{C}^-$. Moreover,

- if $u_1^T v_1 > u_2^T v_2$, then $\lambda_n = 0$ and $\lambda_{n+1} < 0$ are simple eigenvalues;
- if $u_1^T v_1 < u_2^T v_2$, then $\lambda_n > 0$ and $\lambda_{n+1} = 0$ are simple eigenvalues;
- if $u_1^T v_1 = u_2^T v_2$, then $\lambda_n = \lambda_{n+1} = 0$ is a double eigenvalue with only one linearly independent eigenvector.

Therefore, in all cases, there is a unique invariant subspace of $L$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$. Let the invariant subspace be Im $[U_1^T\ U_2^T]^T$, where $U_1 \in \mathbb{C}^{n\times n}$, $U_2 \in \mathbb{C}^{m\times n}$ and Im $U$ denotes the image (or range) of the matrix $U$. Then $U_1$ is nonsingular and $S = U_2U_1^{-1}$ (see [10]).

When $M$ is an irreducible $M$-matrix, the matrices $D - CS$ and $A - SC$ are also irreducible $M$-matrices by Theorem 1.1. Since $A - SC$ and $(D - CS)^T$ can be written in the form $sI - N$, where $N \geq 0$ is irreducible, it follows from the Perron–Frobenius theorem that there exist unique positive vectors $a$ and $b$ with unit 1-norm such that

$$(A - SC)a = -\lambda_{n+1}a, \quad b^T(D - CS) = \lambda_n b^T. \tag{3.3}$$

Since $M$ is irreducible, we have $C \neq 0$ and thus $b^TCa > 0$. We will need the following result [7] in the perturbation analysis below.

THEOREM 3.1. [illegible] $M$ [illegible] $M$ [illegible] $M$ [illegible] $u_1^Tv_1 \neq u_2^Tv_2$ [illegible] $S_+$ [illegible] (1.1) [illegible]

$$S_+ = S + \delta ab^T, \tag{3.4}$$

[illegible] $a, b$ [illegible] (3.3) [illegible] $\delta = (\lambda_n - \lambda_{n+1})/b^TCa$ [illegible]

$$\sigma(D - CS_+) = \{\lambda_1, \ldots, \lambda_{n-1}, \lambda_{n+1}\}. \tag{3.5}$$

Let $\mathcal{M}$ and $\mathcal{N}$ be any invariant subspaces of $L$. For any fixed norm $\|\cdot\|$ (for definiteness we use the spectral norm), let $\theta(\mathcal{M}, \mathcal{N})$ be the gap between $\mathcal{M}$ and $\mathcal{N}$, defined by

$$\theta(\mathcal{M}, \mathcal{N}) = \|P_{\mathcal{M}} - P_{\mathcal{N}}\|,$$

where $P_M$ and $P_{\mathcal{N}}$ are the orthogonal projectors on $\mathcal{M}$ and $\mathcal{N}$, respectively, with orthogonality defined by the standard scalar product on $\mathbb{C}^{m+n}$. See [8] or [22] for properties of the gap metric.

We first consider the case where $M$ is a nonsingular $M$-matrix or an irreducible singular $M$-matrix with $u_1^Tv_1 \neq u_2^Tv_2$. In this case, since the eigenvalues $\lambda_1, \ldots, \lambda_n$ are disjoint from the eigenvalues $\lambda_{n+1}, \ldots, \lambda_{n+m}$, the invariant subspace corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$,

$$\mathcal{M} = \text{Im}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \text{Im}\begin{bmatrix} I \\ S \end{bmatrix},$$

is known to be Lipschitz stable [8], i.e., there exist constants $\gamma_1, \epsilon > 0$ such that every matrix $K$ satisfying $\|K - L\| < \epsilon$ has an invariant subspace $\mathcal{N}$ for which $\theta(\mathcal{M}, \mathcal{N}) \leq \gamma_1\|K - L\|$. In particular, every $\widetilde{L} = \text{diag}(I, -I)\widetilde{M}$ with $\|\widetilde{L} - L\| < \epsilon$ has an invariant subspace $\mathcal{N}$ for which $\theta(\mathcal{M}, \mathcal{N}) \leq \gamma_1\|\widetilde{L} - L\|$. Let $\mathcal{N} = \text{Im}[V_1^T\ V_2^T]^T$. Then for $\epsilon$ small enough, $V_1$ is nonsingular and we let $T = V_2V_1^{-1}$. Then for $\|\widetilde{M} - M\| = \|\widetilde{L} - L\| < \epsilon$

$$\theta\left(\text{Im}\begin{bmatrix} I \\ S \end{bmatrix}, \text{Im}\begin{bmatrix} I \\ T \end{bmatrix}\right) \leq \gamma_1\|\widetilde{M} - M\|.$$

Note that there is a constant $\gamma_2 > 0$ such that [8]

$$\gamma_2^{-1}\|T - S\| \leq \theta\left(\text{Im}\begin{bmatrix} I \\ S \end{bmatrix}, \text{Im}\begin{bmatrix} I \\ T \end{bmatrix}\right) \leq \gamma_2\|T - S\|.$$

Thus

$$\|T - S\| \le \gamma_1\gamma_2\|\widetilde{M} - M\|.$$

For $\epsilon$ small enough, we know that the eigenvalues of $\widetilde{D} - \widetilde{C}T$ are individually close to the eigenvalues of $D - CS$, and hence they are the $n$ eigenvalues of $\widetilde{L}$ with the largest real parts. It follows that $T = \widetilde{S}$, the minimal nonnegative solution of (1.1) with $M$ replaced by $\widetilde{M}$.

We have thus proved the following result.

THEOREM 3.2. [illegible] $M$ [illegible] $M$ [illegible] $M$ [illegible] $u_1^T v_1 \neq u_2^T v_2$ [illegible] $\gamma > 0$ [illegible] $\epsilon > 0$ [illegible] $\|\widetilde{S} - S\| \le \gamma\|\widetilde{M} - M\|$ [illegible] $\widetilde{M}$ [illegible] $\|\widetilde{M} - M\| < \epsilon$

We now consider the case where $M$ is an irreducible singular $M$-matrix with $u_1^T v_1 = u_2^T v_2$. Let $q_1, q_2, \ldots, q_{n-1}$ be the eigenvectors and generalized eigenvectors corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ and let $v$ be the eigenvector corresponding to the zero eigenvalue. Now,

$$\mathrm{Im}\begin{bmatrix} I \\ S \end{bmatrix} = \mathrm{Im}[q_1\ q_2\ \ldots\ q_{n-1}] \dotplus \mathrm{Im}[v].$$

As in the previous case, there exist constants $\gamma_1, \epsilon > 0$ such that for any $\widetilde{M}$ with $\|\widetilde{M} - M\| < \epsilon$, $\widetilde{L}$ has an invariant subspace $\mathcal{N}_1$ for which

$$\theta(\mathrm{Im}[q_1\ q_2\ \ldots\ q_{n-1}], \mathcal{N}_1) \le \gamma_1\|\widetilde{M} - M\|.$$

We assume that $\epsilon$ is small enough such that the eigenvalues of $\widetilde{L}$ corresponding to $\mathcal{N}_1$ are the $n-1$ eigenvalues of $\widetilde{L}$ with the largest real parts. Note that when $\widetilde{M}$ is close enough to $M$, $\widetilde{M}$ is also irreducible. We consider two cases: (a) $\widetilde{M}$ is nonsingular and (b) $\widetilde{M}$ is singular.

For case (a), $\widetilde{L}$ has an eigenvalue $\widetilde{\lambda}_n > 0$ that is a perturbation of the zero eigenvalue (with index two) of $L$. The eigenvector $\widetilde{v}$ corresponding to $\widetilde{\lambda}_n$ is such that

$$\theta(\mathrm{Im}[v], \mathrm{Im}[\widetilde{v}]) \le \gamma_2\|\widetilde{M} - M\|^{1/2}$$

for some $\gamma_2 > 0$ (see section 16.5 of [8] or section 5 of [9]). Now, there are constants $\gamma_3, \gamma_4 > 0$ such that [8]

$$\begin{aligned}
&\theta(\mathrm{Im}[q_1\ q_2\ \ldots\ q_{n-1}] \dotplus \mathrm{Im}[v], \mathcal{N}_1 \dotplus \mathrm{Im}[\widetilde{v}]) \\
&\quad \le \gamma_3[\theta(\mathrm{Im}[q_1\ q_2\ \ldots\ q_{n-1}], \mathcal{N}_1) + \theta(\mathrm{Im}[v], \mathrm{Im}[\widetilde{v}])] \\
&\quad \le \gamma_4\|\widetilde{M} - M\|^{1/2}.
\end{aligned}$$

It then follows as before that $\|\widetilde{S} - S\| \le \gamma\|\widetilde{M} - M\|^{1/2}$ for some $\gamma > 0$.

For case (b), let $\widetilde{v}$ be the eigenvector corresponding to the zero eigenvalue of $\widetilde{L}$. Then $v$ and $\widetilde{v}$ are also eigenvectors of $M$ and $\widetilde{M}$ corresponding to its simple zero eigenvalue. It is known that

$$\theta(\mathrm{Im}[v], \mathrm{Im}[\widetilde{v}]) \le \gamma_2\|\widetilde{M} - M\|$$

for some $\gamma_2 > 0$. If $0 = \widetilde{\lambda}_n \geq \widetilde{\lambda}_{n+1}$ then as before $\|\widetilde{S} - S\| \leq \gamma \|\widetilde{M} - M\|$ for some $\gamma > 0$. If $0 = \widetilde{\lambda}_{n+1} < \widetilde{\lambda}_n$ then we use Theorem 3.1 with $M$ replaced by $\widetilde{M}$ (so accordingly we have $\widetilde{S}, \widetilde{S}_+, \widetilde{a}, \widetilde{b}$, etc.) to get

$$\|\widetilde{S}_+ - S\| \leq \gamma_3 \|\widetilde{M} - M\|$$

for some $\gamma_3 > 0$. Note that $\|\widetilde{S}_+ - \widetilde{S}\| \leq \|\widetilde{\delta}\ \widetilde{a}\ \widetilde{b}^T\| \leq \gamma_4 |\widetilde{\lambda}_n|$ for some $\gamma_4 > 0$. The eigenvalues of $\widetilde{A} - \widetilde{S}_+\widetilde{C}$ are $-\widetilde{\lambda}_n, -\widetilde{\lambda}_{n+2}, \ldots, -\widetilde{\lambda}_{n+m}$. The simple eigenvalue $-\widetilde{\lambda}_n$ of $\widetilde{A} - \widetilde{S}_+\widetilde{C}$ is a perturbation of the simple eigenvalue $-\lambda_{n+1} = 0$ of $A - SC$. Thus $|\widetilde{\lambda}_n| \leq \gamma_5 \|(\widetilde{A} - \widetilde{S}_+\widetilde{C}) - (A - SC)\| \leq \gamma_6 \|\widetilde{M} - M\|$ for some $\gamma_5, \gamma_6 > 0$. Therefore $\|\widetilde{S} - S\| \leq \|\widetilde{S}_+ - S\| + \|\widetilde{S}_+ - \widetilde{S}\| \leq \gamma \|\widetilde{M} - M\|$ for some $\gamma > 0$.

In summary, we have shown the following.

THEOREM 3.3. [illegible] $M$ [illegible] $M$ [illegible] $u_1^T v_1 = u_2^T v_2$ [illegible] $\gamma > 0$ [illegible] $\epsilon > 0$ [illegible]

(a) $\|\widetilde{S} - S\| \leq \gamma \|\widetilde{M} - M\|^{1/2}$ [illegible] $\widetilde{M}$ [illegible] $\|\widetilde{M} - M\| < \epsilon$.

(b) $\|\widetilde{S} - S\| \leq \gamma \|\widetilde{M} - M\|$ [illegible] singular $\widetilde{M}$ [illegible] $\|\widetilde{M} - M\| < \epsilon$

We illustrate the results in Theorem 3.3 with a simple example.

Consider the matrix

$$M = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

and its three different perturbations

$$M_1 = \begin{bmatrix} 1+\epsilon & -1 \\ -1 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & -(1+\epsilon) \\ -1 & 1+\epsilon \end{bmatrix}, \quad M_3 = \begin{bmatrix} 1 & -1 \\ -(1+\epsilon) & 1 \end{bmatrix},$$

where $0 < \epsilon < 1$. Note that $M$ satisfies the condition in Theorem 3.3, and that $S = 1$ for the corresponding NARE (1.1). $M_1$ is a nonsingular $M$-matrix and the corresponding minimal solution is $S_1 = \frac{1}{2}(2 + \epsilon - \sqrt{4\epsilon + \epsilon^2}) \sim 1 - \epsilon^{1/2}$, which is the situation in Theorem 3.3(a). $M_2$ is an irreducible singular $M$-matrix and the corresponding minimal solution is $S_2 = 1/(1+\epsilon) \sim 1 - \epsilon$, which is the situation in Theorem 3.3(b). $M_3$ is not an $M$-matrix and the corresponding NARE does not have real solutions.

The continuity of the minimal solution shown in Theorem 3.3 can be used to prove the next result, where the statements are stronger than those given in [10, Thm. 4.8]. The result will be needed in section 4.

THEOREM 3.4. [illegible] $M$ [illegible] $M$ [illegible]

(a) [illegible] $u_1^T v_1 = u_2^T v_2$ [illegible] $Sv_1 = v_2$ [illegible] $\widehat{S}v_2 = v_1$

(b) [illegible] $u_1^T v_1 > u_2^T v_2$ [illegible] $Sv_1 = v_2$ [illegible] $\widehat{S}v_2 < v_1$

(c) [illegible] $u_1^T v_1 < u_2^T v_2$ [illegible] $Sv_1 < v_2$ [illegible] $\widehat{S}v_2 = v_1$

[illegible] We only need to prove the result for $S$ since the result for $\widehat{S}$ follows immediately by duality. So we need to show $Sv_1 = v_2$ when $u_1^T v_1 \geq u_2^T v_2$ and $Sv_1 < v_2$ when $u_1^T v_1 < u_2^T v_2$. In fact,

$$\begin{aligned}(A - SC)(v_2 - Sv_1) &= Av_2 - SCv_2 + (SCS - AS)v_1 \\ &= Bv_1 - SDv_1 + (SD - B)v_1 = 0.\end{aligned}$$

If $u_1^T v_1 > u_2^T v_2$, then $A - SC$ is nonsingular and so $Sv_1 = v_2$. If $u_1^T v_1 < u_2^T v_2$, then $A - SC$ is an irreducible singular $M$-matrix and $v_2 - Sv_1 \geq 0$ is an eigenvector corresponding to the zero eigenvalue (it is already proved in [10] that $Sv_1 \le v_2$ and $Sv_1 \ne v_2$). By the Perron–Frobenius theorem, $v_2 - Sv_1 > 0$ and so $Sv_1 < v_2$. If $u_1^T v_1 = u_2^T v_2$, then for

$$M(\alpha) = \begin{bmatrix} D & -C \\ -\alpha B & \alpha A \end{bmatrix}$$

with $\alpha > 1$, we have

$$u_1(\alpha) = u_1, \quad u_2(\alpha) = \alpha^{-1} u_2, \quad v_1(\alpha) = v_1, \quad v_2(\alpha) = v_2.$$

So we have $u_1(\alpha)^T v_1(\alpha) > u_2(\alpha)^T v_2(\alpha)$. It follows that $S(\alpha)v_1(\alpha) = v_2(\alpha)$. However, $\lim_{\alpha\to 1^+} S(\alpha) = S$ by Theorem 3.3 and so $Sv_1 = v_2$. ☐

## 4. Applicability of the shifted LR algorithm.

In this section we assume that $M$ is an irreducible singular $M$-matrix. For the NARE (1.1) arising in the study of Markov models, we have $Me = 0$, where $e$ is the vector of ones. In that case, we may take $v_1 = e \in \mathbb{R}^n$ and $v_2 = e \in \mathbb{R}^m$ in (1.3).

If $M$ is a general irreducible singular $M$-matrix, we can transform (1.1) into a new equation for which $v_1 = e$ and $v_2 = e$. More precisely, (1.1) can be rewritten as

$$W(V_1^{-1}CV_2)W - W(V_1^{-1}DV_1) - (V_2^{-1}AV_2)W + V_2^{-1}BV_1 = 0 \tag{4.1}$$

with $V_1 = \operatorname{diag}(v_1)$, $V_2 = \operatorname{diag}(v_2)$, and $W = V_2^{-1}XV_1$. Note that the minimal nonnegative solution of (4.1) is $\overline{S} = V_2^{-1}SV_1$ and that

$$\begin{bmatrix} V_1^{-1}DV_1 & -V_1^{-1}CV_2 \\ -V_2^{-1}BV_1 & V_2^{-1}AV_2 \end{bmatrix} \begin{bmatrix} e \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.2}$$

It is clear that the leftmost matrix in (4.2) is still an irreducible singular $M$-matrix. From now on, we assume that $M$ is an irreducible singular $M$-matrix with $Me = 0$.

Ramaswami [26] made the interesting observation that the matrix equation (1.1) is closely related to a quadratic matrix equation arising in quasi-birth-death processes. To see this connection, let

$$a_* = \max_{1\le i\le m} a_{ii}, \quad d_* = \max_{1\le i\le n} d_{ii}, \quad \theta_* = \max(a_*, d_*). \tag{4.3}$$

Choose a number $\theta \ge \theta_*$ and let $P = I - \frac{1}{\theta}M$. Then $P$ is nonnegative with $Pe = e$, i.e., $P$ is a stochastic matrix. Let

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix},$$

where the partitioning is conformable with that for the matrix $M$. Thus

$$P_{11} = I - \frac{1}{\theta}D, \quad P_{12} = \frac{1}{\theta}C, \quad P_{21} = \frac{1}{\theta}B, \quad P_{22} = I - \frac{1}{\theta}A. \tag{4.4}$$

Ramaswami [26] constructed three nonnegative matrices from $P$:

$$A_0 = \begin{bmatrix} P_{11} & 0 \\ \frac{1}{2}P_{21} & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & P_{12} \\ 0 & \frac{1}{2}P_{22} \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2}I \end{bmatrix}. \tag{4.5}$$

Associated with the matrices $A_0, A_1, A_2$ are the matrix equation

$$G = A_0 + A_1 G + A_2 G^2 \tag{4.6}$$

and its dual equation

$$F = A_2 + A_1 F + A_0 F^2. \tag{4.7}$$

We let $G$ and $F$ be the minimal nonnegative solutions of (4.6) and (4.7), respectively.

The next two results are known (see [26, Thm. 4.1] and [14]).

Proposition 4.1. [illegible] (4.6) [illegible]

$$G = \begin{bmatrix} P_{11} + P_{12}S & 0 \\ S & 0 \end{bmatrix},$$

[illegible] $S$ [illegible] (1.1)

Proposition 4.2. [illegible] (4.7) [illegible]

$$F = \begin{bmatrix} 0 & \widehat{S} \\ 0 & (2I - P_{22} - P_{21}\widehat{S})^{-1} \end{bmatrix},$$

[illegible] $\widehat{S}$ [illegible] (1.4)

Since $(2I - P_{22} - P_{21}\widehat{S})^{-1} = (I + \frac{1}{\theta}(A - B\widehat{S}))^{-1}$ is a nonnegative matrix, $\rho(F) = \rho((2I - P_{22} - P_{21}\widehat{S})^{-1})$ is the largest positive eigenvalue of $(I + \frac{1}{\theta}(A - B\widehat{S}))^{-1}$, which is $1/(1 - \frac{1}{\theta}\lambda_{n+1})$. Similarly, $\rho(G) = \rho(P_{11} + P_{12}S) = \rho(I - \frac{1}{\theta}(D - CS)) = 1 - \frac{1}{\theta}\lambda_n$.

The solution $G$ can be computed by the LR algorithm [23], which is essentially the cyclic reduction algorithm combined with block-diagonal scaling (see [12]).

Algorithm 4.3. [illegible]

$$L^{(0)} = (I - A_1)^{-1} A_0,$$
$$H^{(0)} = (I - A_1)^{-1} A_2,$$
$$G^{(0)} = L^{(0)},$$
$$T^{(0)} = H^{(0)}.$$

[illegible] $k = 0, 1, \ldots$ [illegible]

$$U^{(k)} = H^{(k)} L^{(k)} + L^{(k)} H^{(k)},$$
$$L^{(k+1)} = (I - U^{(k)})^{-1} (L^{(k)})^2,$$
$$H^{(k+1)} = (I - U^{(k)})^{-1} (H^{(k)})^2,$$
$$G^{(k+1)} = G^{(k)} + T^{(k)} L^{(k+1)},$$
$$T^{(k+1)} = T^{(k)} H^{(k+1)}.$$

It is shown in [23] that the matrices $H^{(k)}$ and $L^{(k)}$ are well defined and nonnegative and that the sequence $\{G^{(k)}\}$ converges quadratically to the matrix $G$, except for a critical case which corresponds to the case $u_1^T e = u_2^T e$ in the NARE (1.1). In the latter case, the convergence is expected to be linear with rate $1/2$ (see [12] and [14]).

When $m = n$, the LR algorithm needs about $\frac{400}{3}n^3$ flops each iteration. Using the special structure of the matrices $A_0, A_1, A_2$, we can simplify the LR algorithm and the simplified algorithm requires about $\frac{124}{3}n^3$ flops each iteration [14]. The simplified LR algorithm is less expensive than Newton's method, which requires roughly $60n^3$ flops each iteration when $m = n$. However, there are examples [1] for which the (simplified) LR algorithm requires many more iterations than Newton's method, even though they both have quadratic convergence.

The matrix $G^{(k)}$ from Algorithm 4.3 has the form

$$G^{(k)} = \begin{bmatrix} G_1^{(k)} & 0 \\ G_2^{(k)} & 0 \end{bmatrix},$$

and the solution $S$ is approximated by the matrices $S_k = G_2^{(k)}$. It is shown in [14] that

$$\limsup_{k\to\infty} \sqrt[2^{k+1}]{\|S_k - S\|} \le \rho(F)\rho(G), \tag{4.8}$$

so $S_k$ converges to $S$ quadratically when $\rho(F)\rho(G) < 1$ and the convergence will be fast if $\rho(F)\rho(G)$ is not close to 1.

Since

$$\rho(F) = 1/\left(1 - \frac{1}{\theta}\lambda_{n+1}\right), \quad \rho(G) = 1 - \frac{1}{\theta}\lambda_n$$

are nondecreasing functions of $\theta$ for $\theta \ge \theta_*$, we should take $\theta = \theta_*$ in (4.4) to have faster convergence for the (simplified) LR algorithm.

Note that when $u_1^T e = u_2^T e$, $Se = e$ and $\widehat{S}e = e$ by Theorem 3.4. So $Fe = Ge = e$, $\rho(F) = \rho(G) = 1$ and the convergence is expected to be linear with rate 1/2. To have faster convergence when $u_1^T e \ge u_2^T e$, we need to use a shift technique [19] for the (simplified) LR algorithm. The case $u_1^T e < u_2^T e$ for the NARE will be reduced to the case $u_1^T e > u_2^T e$ for a new NARE of the same type.

**4.1. Case $u_1^T e \ge u_2^T e$.** In this subsection we assume $u_1^T e \ge u_2^T e$. In this case $Se = e$ and so $G$ is stochastic. It is shown in [14] that the only eigenvalue of $G$ on the unit circle is the simple eigenvalue 1.

The shift technique introduced in [19] is $H = G - ev^T$, where $v > 0$ and $v^T e = 1$. For our purposes here, we only require that $v \ge 0$ and $v^T e = 1$. Then the eigenvalues of $H$ are those of $G$ except that the eigenvalue 1 of $G$ is replaced by 0, and $H$ is a solution of the new equation

$$H = B_0 + B_1 H + B_2 H^2, \tag{4.9}$$

where

$$B_0 = A_0(I - ev^T), \quad B_1 = A_1 + A_2 ev^T, \quad B_2 = A_2. \tag{4.10}$$

It is shown in [14] that there is a matrix $K$ with $\rho(K) = \rho(F)$ such that

$$K = B_2 + B_1 K + B_0 K^2. \tag{4.11}$$

To find the solution $H$ of (4.9), we can apply Algorithm 4.3 with the triple $(A_0, A_1, A_2)$ replaced by the triple $(B_0, B_1, B_2)$. To avoid confusion, we will put a "hat" on each sequence generated. We take

$$v = \begin{bmatrix} p \\ 0 \end{bmatrix}, \tag{4.12}$$

where $p \in \mathbb{R}^n$ is positive and $p^T e = 1$. In this way we can get a simplified LR algorithm as before, with no increase in computational work for each iteration. Note that $S$ is now approximated by $\widehat{S}_k = \widehat{G}_2^{(k)} + ep^T$.

It is shown in [14] that when Algorithm 4.3 is applied with $(A_0, A_1, A_2)$ replaced by $(B_0, B_1, B_2)$, the matrix $I - B_1$ in the initialization step is always invertible. Assuming that $I - \widehat{U}^{(k)}$ is invertible for each $k \geq 0$, it is shown in [14] that

$$\limsup_{k\to\infty} \sqrt[2^{k+1}]{\|\widehat{S}^{(k)} - S\|} \leq \rho(K)\rho(H) = \rho(F)\rho(H) < 1. \tag{4.13}$$

Since $\rho(H) < \rho(G)$, the shift technique has improved the speed of convergence. In particular, $\widehat{S}^{(k)}$ converges to $S$ quadratically whenever $u_1^T e \geq u_2^T e$. It is also shown in [14] that $I - \widehat{U}^{(k)}$ converges to $I$ quadratically, assuming that $I - \widehat{U}^{(k)}$ is nonsingular for all $k \geq 0$.

The problem as to whether the matrices $I - \widehat{U}^{(k)}$ could be singular for small $k$ was unsolved in [14]. We will now solve this problem.

We proceed as in [6] but depart from [6] at some point. Let

$$T_k = \begin{bmatrix} I - A_1 & -A_2 & & \\ -A_0 & I - A_1 & \ddots & \\ & \ddots & \ddots & -A_2 \\ & & -A_0 & I - A_1 \end{bmatrix}$$

and

$$\widehat{T}_k = \begin{bmatrix} I - B_1 & -B_2 & & \\ -B_0 & I - B_1 & \ddots & \\ & \ddots & \ddots & -B_2 \\ & & -B_0 & I - B_1 \end{bmatrix}$$

be block $k \times k$ Toeplitz matrices. Since the LR algorithm is well defined if and only if the cyclic reduction (CR) algorithm is well defined [5], it follows from Theorem 13 of [3] that the matrices $T_{2^j-1}$ are nonsingular for all $j \geq 1$ and that $I - \widehat{U}^{(k)}$ are nonsingular for all $k \geq 0$ if $\widehat{T}_{2^j-1}$ are nonsingular for all $j \geq 2$. The relation between $T_k$ and $\widehat{T}_k$ (for $k \geq 3$) has been obtained in [6] as

$$\widehat{T}_k = T_k \begin{bmatrix} I & & & \\ V & I & & \\ \vdots & \ddots & \ddots & \\ V & \dots & V & I \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -A_2 \end{bmatrix} \begin{bmatrix} V & V & \dots & V \end{bmatrix}, \tag{4.14}$$

where $V = ev^T$. Note that this relation can be obtained directly from (4.10). Let $Q_k$ and $P_k$ be the $(k, 1)$ block and $(k, k)$ block of $T_k^{-1}$, respectively. From (4.14), it is shown in [6] that $\widehat{T}_k$ is nonsingular if and only if $v^T P_k A_2 e \neq 1$. From the proof of Theorem 9 in [6] we also know that

$$v^T Q_k A_0 e + v^T P_k A_2 e = 1. \tag{4.15}$$

In the case where $v$ is taken to be positive and $u_1^T e > u_2^T e$, it has been shown in [6] that $v^T P_k A_2 e \neq 1$, using, among other things, the canonical factorizations of matrix polynomials and the so-called asymptotic applicability of the SCR (CR with a shift technique). So, the argument in [6] is very involved and it does not cover the case $u_1^T e = u_2^T e$. Suppose SCR were to break down for the case $u_1^T e = u_2^T e$. Then near-breakdown would happen to SCR with $u_1^T e > u_2^T e$, but $u_1^T e \approx u_2^T e$. Moreover, as we mentioned earlier, we need to take the vector $v$ in the form (4.12) to avoid an increase in computational work when using the shift technique. Fortunately, we can prove the applicability of the LR algorithm, with a shift given by (4.12), for all cases with $u_1^T e \geq u_2^T e$ and $\theta > \theta_*$. Moreover, the proof is very simple.

In fact, what we need to prove is $v^T Q_k A_0 e > 0$, which implies $v^T P_k A_2 e \neq 1$ by (4.15). Note that

$$T_k^{-1} \geq \begin{bmatrix} I & & & \\ -A_0 & I & & \\ & \ddots & \ddots & \\ & & -A_0 & I \end{bmatrix}^{-1} = \begin{bmatrix} I & & & \\ A_0 & I & & \\ \vdots & \ddots & \ddots & \\ A_0^{k-1} & \cdots & A_0 & I \end{bmatrix}. \tag{4.16}$$

So $Q_k \geq A_0^{k-1}$ and hence $v^T Q_k A_0 e \geq v^T A_0^k e$. For $A_0$ given by (4.5), we have

$$A_0^k = \begin{bmatrix} P_{11}^k & 0 \\ \frac{1}{2} P_{21} P_{11}^{k-1} & 0 \end{bmatrix}.$$

Therefore, $v^T Q_k A_0 e \geq p^T P_{11}^k e$, by (4.12). Recall that the nonnegative matrix $P_{11}$ is given by $P_{11} = I - \frac{1}{\theta} D$. If the diagonal elements $d_{ii}$ of $D$ are not all equal or $a_*$ and $d_*$ defined in (4.3) satisfy $d_* < a_*$, then $P_{11}$ has at least one nonzero diagonal element and hence $p^T P_{11}^k e > 0$ for all $k \geq 1$ and for all $\theta \geq \theta_*$. If the elements $d_{ii}$ are all equal and $d_* \geq a_*$, then $p^T P_{11}^k e > 0$ for all $k \geq 1$ and all $\theta > \theta_* = d_*$.

THEOREM 4.4. [illegible] 4.3 [illegible] $A_0, A_1, A_2$ [illegible] (4.5) [illegible] $B_0, B_1, B_2$ [illegible] (4.10) [illegible] $\theta \geq \theta_*$ [illegible] $d_{ii}$ [illegible] $D$ [illegible] $d_* < a_*$ [illegible] $\theta > \theta_*$ [illegible] $d_{ii}$ [illegible] $d_* \geq a_*$

When the elements $d_{ii}$ of $D$ are all equal, it is possible for $P_{11}$ to be nilpotent if we take $\theta = \theta_*$. One simple example is

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix}. \tag{4.17}$$

For this example with $\theta = 1$, $p^T P_{11}^k e = 0$ for $k \geq 2$. However, it is very likely that we still have $v^T Q_k A_0 e > 0$ since the lower bound in (4.16) is not tight.

For the LR algorithm without a shift, the number $\rho(F)\rho(G)$ in (4.8) in minimized for $\theta = \theta_*$. So $\theta = \theta_*$ is optimal in this sense and should be recommended. For the LR algorithm with a shift, however, the optimal $\theta$ should minimize $\rho(F)\rho(H)$ in (4.13).

When $u_1^Te = u_2^Te$, we have $\lambda_{n+1} = 0$ and $\rho(F) = 1$ for any $\theta$. When $u_1^Te > u_2^Te$ but $u_1^Te \approx u_2^Te$, we have $\lambda_{n+1} \approx 0$ and hence the effect of $\theta$ on $\rho(F)$ is very limited. So one should try to minimize $\rho(H)$. Note that $\rho(H) = \max_{1\le i\le n-1} |1-\frac{1}{\theta}\lambda_i|$. For the matrix $M$ given by (4.17), the corresponding matrix $L$ has eigenvalues $\sqrt{2}, 0, 0, -\sqrt{2}$. So $\rho(F) = 1$ and $\rho(H)$ is minimized for $\theta = \sqrt{2}$ and the minimum is 0. This example shows that $\theta = \theta_*$ is not necessarily optimal when the shift technique is used. We can also give a necessary and sufficient condition for $\theta_*$ to be optimal. Let $D = \{z \in \mathbb{C} : |z-1| < 1\}$. Then $\lambda_i/\theta_* \in D$ for $i = 1, \ldots, n-1$ since $\rho(H) < 1$. Let $D_1 = \{z \in \mathbb{C} : |z-1/2| \le 1/2\}$, $D_2 = D \setminus D_1$, $I_1 = \{1 \le i \le n-1 : \lambda_i/\theta_* \in D_1\}$, and $I_2 = \{1 \le i \le n-1 : \lambda_i/\theta_* \in D_2\}$. Then we have the following result.

PROPOSITION 4.5. [illegible] $\theta \in [\theta_*, \infty)$ $\rho(H)$ [illegible] $\theta = \theta_*$ [illegible]

$$\max_{i\in I_1} |1-\lambda_i/\theta_*| \ge \max_{i\in I_2} |1-\lambda_i/\theta_*|,$$

[illegible]

[illegible]. Note that for any point (other than 0) on the circle $|z-1/2| = 1/2$, the boundary of $D_1$, the line passing through $z$ and 0 is perpendicular to the line passing through $z$ and 1. If $\max_{i\in I_1} |1-\lambda_i/\theta_*| \ge \max_{i\in I_2} |1-\lambda_i/\theta_*|$, then for any $\theta > \theta_*$ and $i \in I_1$, which is nonempty, $|1-\lambda_i/\theta| > |1-\lambda_i/\theta_*|$ and thus $\rho(H)$ is minimized at $\theta_*$. On the other hand, if $\max_{i\in I_1} |1-\lambda_i/\theta_*| < \max_{i\in I_2} |1-\lambda_i/\theta_*|$, we can take $\theta > \theta_*$ such that

$$\max_{i\in I_1} |1-\lambda_i/\theta| < \max_{i\in I_2} |1-\lambda_i/\theta| < \max_{i\in I_2} |1-\lambda_i/\theta_*|.$$

(The first inequality holds when $\theta-\theta_*$ is small enough and the second inequality holds when $\theta-\theta_*$ is small enough so that $\lambda_i/\theta \in D_2$ for $i \in I_2$.) Thus $\rho(H)$ does not attain its minimum at $\theta_*$. $\square$

In practice, we would not compute the eigenvalues $\lambda_1, \ldots, \lambda_{n-1}$ when we use the LR algorithm. However, the above result shows that $\theta = \theta_*$ is often not optimal when the shift technique is used. Therefore, when the diagonal elements $d_{ii}$ of $D$ are all equal and $d_* \ge a_*$, we can simply take $\theta > \theta_* = d_*$ (say $\theta = 1.1\theta_*$) to ensure the applicability of the LR algorithm with a shift.

**4.2. Case $u_1^Te < u_2^Te$.** We now assume $u_1^Te < u_2^Te$. Then $Se < e$ by Theorem 3.4. We will reduce this case to the case $u_1^Te > u_2^Te$ for a new NARE of the same type, and the substochastic minimal solution $S$ of the original NARE will be obtained from the stochastic minimal solution of the new NARE. This reduction process is in essence similar to the one given in [25]. The difference is that the reduction here is given directly on the Riccati equation, rather than on the unilateral matrix equation obtained through the Ramaswami construction.

As in [15, Lem. 5.1] we note that the minimal nonnegative solution $S$ of the NARE (1.1) is such that $S = Z^T$, where $Z$ is the minimal nonnegative solution of the new NARE

$$ZC^TZ - ZA^T - D^TZ + B^T = 0. \tag{4.18}$$

As at the beginning of section 4, (4.18) can be rewritten as

$$W(U_2^{-1}C^TU_1)W - W(U_2^{-1}A^TU_2) - (U_1^{-1}D^TU_1)W + U_1^{-1}B^TU_2 = 0, \tag{4.19}$$

with $U_1 = \operatorname{diag}(u_1), U_2 = \operatorname{diag}(u_2)$, and $W = U_1^{-1} Z U_2$. Now the irreducible singular $M$-matrix corresponding to (4.19) is

$$\widehat{M} = \begin{bmatrix} U_2^{-1} A^T U_2 & -U_2^{-1} C^T U_1 \\ -U_1^{-1} B^T U_2 & U_1^{-1} D^T U_1 \end{bmatrix}.$$

It is easy to see that $(u_2^T \ u_1^T)\widehat{M} = 0$ and $\widehat{M}e = 0$. Since $u_2^T e > u_1^T e$, the new NARE (4.19) has a stochastic minimal solution $W$ and it can be computed as in section 4.1. The substochastic minimal solution $S$ of the original NARE is obtained through $S = U_2^{-1} W^T U_1$.

For the above procedure, we need to compute the vector $(u_1^T \ u_2^T)^T$ accurately since it determines the coefficient matrices of the NARE (4.19). This can be done by using the LU factorization of the irreducible singular $M$-matrix $M^T$, and the computational work is very minor compared with that required by each iteration for the simplified LR algorithm. So the shift technique is worthwhile as long as we can save one iteration. Moreover, as our perturbation analysis in section 3 suggests, the minimal solution computed by the LR algorithm without a shift is much more vulnerable to rounding errors when $u_1^T e \approx u_2^T e$.

We use one example to illustrate the usefulness of the above procedure. Consider the NARE (1.1) with $m = n = 100$ and

$$A = \begin{bmatrix} 3 & -1 & & \\ & \ddots & \ddots & \\ & & 3 & -1 \\ -1 & & & 1.9 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \\ & & & 0.9 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 2 & -1 & & \\ & 3 & \ddots & \\ & & \ddots & -1 \\ -1 & & & 3 \end{bmatrix}.$$

It is easily verified that $Me = 0$ and $u_1^T e < u_2^T e$. We apply the (simplified) LR algorithm with a shift to the NARE (4.19) (so the matrices $A, B, C, D$ in (4.4) are replaced accordingly), with $\theta = 3$ in (4.4) and $p = m^{-1}e$ in (4.12). After 6 iterations we find an approximation $\widetilde{W}$ to $W$ with $\|\mathcal{R}(\widetilde{W})\|_\infty = 4.4 \times 10^{-11}$. We then use $\widetilde{W}$ to get an approximation $\widetilde{S}$ to $S$ with $\|\mathcal{R}(\widetilde{S})\|_\infty = 6.1 \times 10^{-11}$. A very accurate approximation to $S$ (with residual $2.3 \times 10^{-14}$) can be obtained by performing 7 iterations instead and we take it as the "exact" solution $S$. We now apply the (simplified) LR algorithm without a shift to the NARE (1.1), with $\theta = 3$ in (4.4). We find after 13 iterations an approximation $\widetilde{S}'$ to $S$, with $\|\mathcal{R}(\widetilde{S}')\|_\infty = 6.0 \times 10^{-10}$. However, the accuracy in this case is much lower than the residual suggests. Indeed, we find $\|\widetilde{S} - S\|_\infty = 1.4 \times 10^{-10}$ but $\|\widetilde{S}' - S\|_\infty = 4.2 \times 10^{-7}$. So the (simplified) LR algorithm with a shift is more efficient and more accurate.

**5. Conclusions.** In this further study of a class of NAREs, we have been able to relax the condition for the convergence of Newton's method to the minimal solution. The qualitative perturbation analysis for the minimal solution, while of independent interest, is instructive in designing algorithms for finding more accurate approximations. For the NAREs arising in Markov models, we have shown that the LR algorithm, combined with a shift technique, is breakdown-free in all cases and therefore is guaranteed to find the minimal solution more efficiently and more accurately.

**Acknowledgments.** This work was carried out while the first author visited MIMS in the School of Mathematics at the University of Manchester; he thanks the School for its hospitality. Both authors thank the referees for their helpful comments. In particular, comments from one referee led to a significant improvement of section 4.2.

## REFERENCES

[1] N. G. Bean, M. M. O'Reilly, and P. G. Taylor, *Algorithms for return probabilities for stochastic fluid flows*, Stoch. Models, 21 (2005), pp. 149–184.

[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, revised reprint of the 1979 Academic Press original, SIAM, Philadelphia, PA, 1994.

[3] D. A. Bini, L. Gemignani, and B. Meini, *Computations with infinite Toeplitz matrices and polynomials*, Linear Algebra Appl., 343–344 (2002), pp. 21–61.

[4] D. A. Bini, B. Iannazzo, G. Latouche, and B. Meini, *On the solution of Riccati equations arising in fluid queues*, Linear Algebra Appl., 413 (2006), pp. 474–494.

[5] D. A. Bini, G. Latouche, and B. Meini, *Solving matrix polynomial equations arising in queueing problems*, Linear Algebra Appl., 340 (2002), pp. 225–244.

[6] D. A. Bini, B. Meini, and I. M. Spitkovsky, *Shift techniques and canonical factorizations in the solution of $M/G/1$-type Markov chains*, Stoch. Models, 21 (2005), pp. 279–302.

[7] S. Fital and C.-H. Guo, *Convergence of the solution of a nonsymmetric matrix Riccati differential equation to its stable equilibrium solution*, J. Math. Anal. Appl., 318 (2006), pp. 648–657.

[8] I. Gohberg, P. Lancaster, and L. Rodman, *Invariant Subspaces of Matrices with Applications*, John Wiley & Sons, New York, 1986.

[9] I. Gohberg and L. Rodman, *On the distance between lattices of invariant subspaces of matrices*, Linear Algebra Appl., 76 (1986), pp. 85–120.

[10] C.-H. Guo, *Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for $M$-matrices*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 225–242.

[11] C.-H. Guo, *A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation*, Linear Algebra Appl., 357 (2002), pp. 299–302.

[12] C.-H. Guo, *Convergence analysis of the Latouche–Ramaswami algorithm for null recurrent quasi-birth-death processes*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 744–760.

[13] C.-H. Guo, *On a quadratic matrix equation associated with an $M$-matrix*, IMA J. Numer. Anal., 23 (2003), pp. 11–27.

[14] C.-H. Guo, *Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models*, J. Comput. Appl. Math., 192 (2006), pp. 353–373.

[15] C.-H. Guo, B. Iannazzo, and B. Meini, *On the Doubling Algorithm for a (Shifted) Nonsymmetric Algebraic Riccati Equation*, Technical Report, 2006; available online at http://www.math.uregina.ca/~chguo/gim06.pdf.

[16] C.-H. Guo and A. J. Laub, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 376–391.

[17] X.-X. Guo and Z.-Z. Bai, *On the minimal nonnegative solution of nonsymmetric algebraic Riccati equation*, J. Comput. Math., 23 (2005), pp. 305–320.

[18] X.-X. Guo, W.-W. Lin, and S.-F. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math., 103 (2006), pp. 393–412.

[19] C. He, B. Meini, and N. H. Rhee, *A shifted cyclic reduction algorithm for quasi-birth-death problems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 673–691.

[20] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.

[21] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 228–243.

[22] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon Press, Oxford, 1995.

[23] G. Latouche and V. Ramaswami, *A logarithmic reduction algorithm for quasi-birth-death processes*, J. Appl. Probab., 30 (1993), pp. 650–674.

[24] L.-Z. Lu, *Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 679–685.

[25] V. Ramaswami, *A duality theorem for the matrix paradigms in queueing theory*, Comm. Statist. Stochastic Models, 6 (1990), pp. 151–161.

[26] V. Ramaswami, *Matrix analytic methods for stochastic fluid flows*, in Proceedings of the 16th International Teletraffic Congress, Elsevier Science B. V., Edinburgh, 1999, pp. 1019–1030.

[27] L. C. G. Rogers, *Fluid models in queueing theory and Wiener–Hopf factorization of Markov chains*, Ann. Appl. Probab., 4 (1994), pp. 390–413.

[28] L. C. G. Rogers and Z. Shi, *Computing the invariant law of a fluid model*, J. Appl. Probab., 31 (1994), pp. 885–896.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 413–433

# A PARTIAL CONDITION NUMBER FOR LINEAR LEAST SQUARES PROBLEMS*

MARIO ARIOLI[†], MARC BABOULIN[‡], AND SERGE GRATTON[‡]

**Abstract.** We consider here the linear least squares problem $\min_{y\in\mathbb{R}^n} \|Ay-b\|_2$, where $b\in\mathbb{R}^m$ and $A\in\mathbb{R}^{m\times n}$ is a matrix of full column rank $n$, and we denote $x$ its solution. We assume that both $A$ and $b$ can be perturbed and that these perturbations are measured using the Frobenius or the spectral norm for $A$ and the Euclidean norm for $b$. In this paper, we are concerned with the condition number of a linear function of $x$ ($L^Tx$, where $L\in\mathbb{R}^{n\times k}$) for which we provide a sharp estimate that lies within a factor $\sqrt{3}$ of the true condition number. Provided the triangular $R$ factor of $A$ from $A^TA=R^TR$ is available, this estimate can be computed in $2kn^2$ flops. We also propose a statistical method that estimates the partial condition number by using the exact condition numbers in random orthogonal directions. If $R$ is available, this statistical approach enables us to obtain a condition estimate at a lower computational cost. In the case of the Frobenius norm, we derive a closed formula for the partial condition number that is based on the singular values and the right singular vectors of the matrix $A$.

**Key words.** linear least squares, normwise condition number, statistical condition estimate, parameter estimation

**AMS subject classifications.** 65F20, 65F35, 15A09, 15A12

**DOI.** 10.1137/050643088

## 1. Introduction.

Perturbation theory has been applied to many problems of linear algebra such as linear systems, linear least squares, or eigenvalue problems [1, 4, 11, 18]. In this paper we consider the problem of calculating the quantity $L^Tx$, where $x$ is the solution of the linear least squares problem (LLSP) $\min_{x\in\mathbb{R}^n} \|Ax-b\|_2$, where $b\in\mathbb{R}^m$ and $A\in\mathbb{R}^{m\times n}$ is a matrix of full column rank $n$. This estimation is a fundamental problem of parameter estimation in the framework of the Gauss–Markov model [17, p. 137]. More precisely, we focus here on the evaluation of the sensitivity of $L^Tx$ to small perturbations of the matrix $A$ and/or the right-hand side $b$, where $L\in\mathbb{R}^{n\times k}$ and $x$ is the solution of the LLSP.

The interest for this question stems, for instance, from parameter estimation where the parameters of the model can often be divided into two parts: the variables of physical significance and a set of ancillary variables involved in the models. For example, this situation occurs in the determination of positions using the GPS system, where the three-dimensional coordinates are the quantities of interest, but the statistical model involves other parameters such as clock drift and GPS ambiguities [12] that are generally estimated during the solution process. It is then crucial to ensure that the solution components of interest can be computed with satisfactory accuracy. The main goal of this paper is to formalize this problem in terms of a condition number and to describe practical methods to compute or estimate this quantity. Note that as far as the sensitivity of a subset of the solution components is concerned, the matrix $L$ is a projection whose columns consist of vectors of the canonical basis of $\mathbb{R}^n$.

---

[*] Received by the editors October 19, 2005; accepted for publication (in revised form) by H. Park September 28, 2006; published electronically April 3, 2007.
http://www.siam.org/journals/simax/29-2/64308.html

[†] Rutherford Appleton Laboratory, Oxfordshire, OX11 0QX, UK (m.arioli@rl.ac.uk).

[‡] CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse Cedex, France (baboulin@cerfacs.fr, gratton@cerfacs.fr).

The condition number of a map $g : \mathbb{R}^m \mapsto \mathbb{R}^n$ at $y_0$ measures the sensitivity of $g(y_0)$ to perturbations of $y_0$. If we assume that the data space $\mathbb{R}^m$ and the solution space $\mathbb{R}^n$ are equipped, respectively, with the norms $\|.\|_{\mathcal{D}}$ and $\|.\|_{\mathcal{S}}$, the condition number $K(y_0)$ is defined by

$$K(y_0) = \lim_{\delta \to 0} \sup_{0 < \|y_0 - y\|_{\mathcal{D}} \leq \delta} \frac{\|g(y_0) - g(y)\|_{\mathcal{S}}}{\|y_0 - y\|_{\mathcal{D}}}, \tag{1.1}$$

whereas the relative condition number is defined by $K^{(rel)}(y_0) = K(y_0)\|y_0\|_{\mathcal{D}} / \|g(y_0)\|_{\mathcal{S}}$. This definition shows that $K(y_0)$ measures an asymptotic sensitivity and that this quantity depends on the chosen norms for the data and solution spaces. If $g$ is a Fréchet-differentiable (F-differentiable) function at $y_0$, then $K(y_0)$ is the norm of the F-derivative $|||g'(y_0)|||$) (see [6]), where $|||.|||$ is the operator norm induced by the choice of the norms on the data and solution spaces.

For the full rank LLSP, we have $g(A,b) = (A^TA)^{-1}A^Tb$. If we consider the product norm $\|(A,b)\|_F = \sqrt{\|A\|_{\mathrm{F}}^2 + \|b\|_2^2}$ for the data space and $\|x\|_2$ for the solution space, then [8] gives an explicit formula for the relative condition number $K^{(rel)}(A,b)$:

$$K^{(rel)}(A,b) = \left\|A^\dagger\right\|_2 \left(\left\|A^\dagger\right\|_2^2 \|r\|_2^2 + \|x\|_2^2 + 1\right)^{\frac{1}{2}} \frac{\|(A,b)\|_F}{\|x\|_2},$$

where $A^\dagger$ denotes the pseudoinverse of $A$, $r = b - Ax$ is the residual vector, and $\|.\|_{\mathrm{F}}$ and $\|.\|_2$ are, respectively, the Frobenius and Euclidean norms. But does the value of $K^{(rel)}(A,b)$ give us useful information about the sensitivity of $L^Tx$? Can it in some cases overestimate the error in components or on the contrary be too optimistic?

Let us consider the following example:

$$A = \begin{pmatrix} 1 & 1 & \epsilon^2 \\ \epsilon & 0 & \epsilon^2 \\ 0 & \epsilon & \epsilon^2 \\ \epsilon^2 & \epsilon^2 & 2 \end{pmatrix}, \quad x = \begin{pmatrix} \epsilon \\ \epsilon \\ \frac{1}{\epsilon} \end{pmatrix}, \quad \text{and} \quad b = \begin{pmatrix} 3\epsilon \\ \epsilon^2 + \epsilon \\ \epsilon^2 + \epsilon \\ 2\epsilon^3 + \frac{2}{\epsilon} \end{pmatrix},$$

where $x$ is the exact solution of the LLSP $\min_{x \in \mathbb{R}^3} \|Ax - b\|_2$. If we take $\epsilon = 10^{-8}$, then we have $x = (10^{-8}, 10^{-8}, 10^8)^T$ and the solution computed in MATLAB using a machine precision $2.22 \cdot 10^{-16}$ is $\tilde{x} = (1.5 \cdot 10^{-8}, 1.5 \cdot 10^{-8}, 10^8)^T$. The LLSP condition number is $K^{(rel)}(A,b) = 2.4 \cdot 10^8$ and the relative errors on the components of $x$ are

$$\frac{|x_1 - \tilde{x}_1|}{|x_1|} = \frac{|x_2 - \tilde{x}_2|}{|x_2|} = 0.5 \quad \text{and} \quad \frac{|x_3 - \tilde{x}_3|}{|x_3|} = 0.$$

Then, if $L = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$, we expect a large value for the condition number of $L^Tx$ because there is a 50% relative error on $x_1$ and $x_2$. If now $L = (0,0,1)^T$, then we expect that the condition number of $L^Tx$ would be close to 1 because $\tilde{x}_3 = x_3$. For these two values of $L$, the LLSP condition number is far from giving a good idea of the sensitivity of $L^Tx$. Note in this case that the perturbations are due to roundoff errors.

Let us now consider a simple example in the framework of parameter estimation where, in addition to roundoff errors, random errors are involved. Let $b = \{b_i\}_{i=1,\ldots,10}$ be a series of observed values depending on data $s = \{s_i\}$, where $s_i = 10 + i, i = 1, \ldots, 10$. We determine a 3-degree polynomial that approximates $b$ in the least squares sense, and we suppose that the following relationship holds:

$$b = x_1 + x_2 \frac{1}{s} + x_3 \frac{1}{s^2} + x_4 \frac{1}{s^3} \quad \text{with } x_1 = x_2 = x_3 = x_4 = 1.$$

We assume that the perturbation on each $b_i$ is $10^{-8}$ multiplied by a normally distributed random number and denote by $\tilde{b} = \{\tilde{b_i}\}_{i=1,\ldots,10}$ the perturbed quantity. This corresponds to the LLSP $\min_{x\in\mathbb{R}^4} \|Ax - \tilde{b}\|_2$, where $A$ is the Vandermonde matrix defined by $A_{ij} = \frac{1}{s_i^{j-1}}$. Let $\tilde{x}$ and $\tilde{y}$ be the computed solutions corresponding to two perturbed right-hand sides. Then we obtain the following relative errors on each component:

$$\frac{|\tilde{x}_1 - \tilde{y}_1|}{|\tilde{x}_1|} = 2 \cdot 10^{-7}, \frac{|\tilde{x}_2 - \tilde{y}_2|}{|\tilde{x}_2|} = 6 \cdot 10^{-6}, \frac{|\tilde{x}_3 - \tilde{y}_3|}{|\tilde{x}_3|} = 6 \cdot 10^{-5}, \text{ and } \frac{|\tilde{x}_4 - \tilde{y}_4|}{|\tilde{x}_4|} = 10^{-4}.$$

We have $K^{(rel)}(A, b) = 3.1 \cdot 10^5$. Regarding the disparity between the sensitivity of each component, we need a quantity that evaluates more precisely the sensitivity of each solution component of the LLSP.

The idea of analyzing the accuracy of some solution components in linear algebra is by no means new. For linear systems $Ax = b$, $A \in \mathbb{R}^n$ and for LLSP, [3] defines so-called componentwise condition numbers that correspond to amplification factors of the relative errors in solution components due to perturbations of data $A$ or $b$ and explains how to estimate them. In our formalism, these quantities are upper bounds of the condition number of $L^T x$, where $L$ is a column of the identity matrix. We also emphasize that the term "componentwise" refers here to the solution components and must be distinguished from the metric used for matrices and for which [21] provides a condition number for generalized inversion and linear least squares.

For LLSP, [14] provides a statistical estimate for componentwise condition numbers due to either relative or structured perturbations. In the case of linear systems, [2] proposes a statistical approach, based on [13] that enables one to compute the condition number of $L^T x$ in $\mathcal{O}(n^2)$.

Our approach differs from the previous studies in the following aspects:

1. We are interested in the condition of $L^T x$, where $L$ is a general matrix and not only a canonical vector of $\mathbb{R}^n$.
2. We are looking for a condition number based on the F-derivative, and not only for an upper bound of this quantity.

We present in this paper three ways to obtain information on the condition of $L^T x$. The first one uses an explicit formula based on the singular value decomposition (SVD) of $A$. The second is at the same time an upper bound of this condition number and a sharp estimate of it. The third method supplies a statistical estimate. The choice between these three methods will depend on the size of the problem (computational cost) and on the accuracy desired for this quantity.

This paper is organized as follows. In section 2, we define the notion of a partial condition number. Then, when perturbations on $A$ are measured using a Frobenius norm, we give a closed formula for this condition number in the general case where $L \in \mathbb{R}^{n\times k}$ and in the particular case when $L \in \mathbb{R}^n$. In section 3, we establish bounds of the partial condition number in Frobenius as well as in spectral norm, and we show that these bounds can be considered as sharp estimates of it. In section 4 we describe a statistical method that enables us to estimate the partial condition number. In section 5 we present numerical results in order to compare the statistical estimate and the exact condition number on sample matrices $A$ and $L$. In section 6 we give a summary comparing the three ways to compute the condition of $L^T x$ as well as a numerical illustration. Finally some concluding remarks are given in section 7.

Throughout this paper we will use the following notation. We use the Frobenius norm $\|.\|_F$ and the spectral norm $\|.\|_2$ on matrices and the usual Euclidean $\|.\|_2$ on vectors. The matrix $I$ is the identity matrix and $e_i$ is the $i$th canonical vector. We also denote by $\mathrm{Im}(A)$ the space spanned by the columns of $A$ and by $\mathrm{Ker}(A)$ the null space of $A$.

**2. The partial condition number of an LLSP.** Let $L$ be an $n \times k$ matrix, with $k \leq n$. We consider the function

$$\begin{array}{rccl} g : & \mathbb{R}^{m\times n} \times \mathbb{R}^m & \longrightarrow & \mathbb{R}^k, \\ & A, b & \longmapsto & g(A,b) = L^T x(A,b) = L^T(A^TA)^{-1}A^Tb. \end{array} \tag{2.1}$$

Since $A$ has full rank $n$, $g$ is continuously F-differentiable in a neighborhood of $(A,b)$ and we denote by $g'$ its F-derivative. Let $\alpha$ and $\beta$ be two positive real numbers. In the present paper we consider the Euclidean norm for the solution space $\mathbb{R}^k$. For the data space $\mathbb{R}^{m\times n} \times \mathbb{R}^m$, we use the product norms defined by

$$\|(A,b)\|_F = \sqrt{\alpha^2 \|A\|_F^2 + \beta^2 \|b\|_2^2}, \quad \alpha, \beta > 0,$$

and

$$\|(A,b)\|_2 = \sqrt{\alpha^2 \|A\|_2^2 + \beta^2 \|b\|_2^2}, \quad \alpha, \beta > 0.$$

These norms are very flexible since they allow us to monitor the perturbations on $A$ and $b$. For instance, large values of $\alpha$ (resp., $\beta$ ) enable us to obtain condition number problems where mainly $b$ (resp., $A$) are perturbed. A more general weighted Frobenius norm $\|(AT, \beta b)\|_{\mathrm{F}}$, where T is a positive diagonal matrix, is sometimes chosen. This is the case, for instance, in [20], which gives an explicit expression for the condition number of rank deficient linear least squares using this norm.

According to [6], the absolute condition numbers of $g$ at the point $(A,b)$ using the two product norms defined above is given by

$$\kappa_{g,F}(A,b) = \max_{(\Delta A, \Delta b)} \frac{\|g'(A,b).(\Delta A, \Delta b)\|_2}{\|(\Delta A, \Delta b)\|_F}$$

and

$$\kappa_{g,2}(A,b) = \max_{(\Delta A, \Delta b)} \frac{\|g'(A,b).(\Delta A, \Delta b)\|_2}{\|(\Delta A, \Delta b)\|_2}.$$

The corresponding relative condition numbers of $g$ at $(A,b)$ are expressed by

$$\kappa_{g,F}^{(rel)}(A,b) = \frac{\kappa_{g,F}(A,b) \ \ \|(A,b)\|_F}{\|g(A,b)\|_2}$$

and

$$\kappa_{g,2}^{(rel)}(A,b) = \frac{\kappa_{g,2}(A,b) \ \ \|(A,b)\|_2}{\|g(A,b)\|_2}.$$

We call the condition numbers related to $L^T x(A,b)$ [illegible] of the LLSP with respect to the linear operator $L$. The partial condition number defined using the product norm $\|(.,.)\|_F$ is given by the following theorem.

THEOREM 1. [illegible] $A = U\Sigma V^T$ [illegible] $A$ [illegible] [7] [illegible] $\Sigma = \mathrm{diag}(\sigma_i)$ [illegible] $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n > 0$ [illegible] $g(A,b) = L^T x(A,b)$ [illegible]

$$\kappa_{g,F}(A,b) = \left\|SV^T L\right\|_2,$$

[illegible] $S \in \mathbb{R}^{n\times n}$ [illegible] $S_{ii} = {\sigma_i}^{-1}\sqrt{\frac{{\sigma_i}^{-2}\|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}$

[illegible] The demonstration is divided into three parts. In Part 1, we establish an explicit formula of $g'(A,b).(\Delta A,\Delta b)$. In Part 2, we derive an upper bound for $\frac{\left\|g'(A,b).(\Delta A,\Delta b)\right\|_2}{\|(\Delta A,\Delta b)\|_F}$. In Part 3, we show that this bound is reached for a particular $(\Delta A,\Delta b)$.

[illegible] 1. Let $\Delta A \in \mathbb{R}^{m\times n}$ and $\Delta b \in \mathbb{R}^m$. Using the chain rules of composition of derivatives, we get

$$\begin{aligned} g'(A,b).(\Delta A,\Delta b) = {} & L^T(A^TA)^{-1}\Delta A^T(b - A(A^TA)^{-1}A^Tb) \\ & - L^T(A^TA)^{-1}A^T\Delta A(A^TA)^{-1}A^Tb + L^TA^\dagger\Delta b, \end{aligned}$$

i.e.,

$$g'(A,b).(\Delta A,\Delta b) = L^T(A^TA)^{-1}\Delta A^T r - L^TA^\dagger\Delta A x + L^TA^\dagger\Delta b. \tag{2.2}$$

We write $\Delta A = \Delta A_1 + \Delta A_2$ by defining $\Delta A_1 = AA^\dagger\Delta A$ (projection of $\Delta A$ onto $\mathrm{Im}(A)$) and $\Delta A_2 = (I - AA^\dagger)\Delta A$ (projection of $\Delta A$ onto $\mathrm{Im}(A)^\perp$). We have $\Delta A_1^T r = 0$ (because $r \in \mathrm{Im}(A)^\perp$) and $A^\dagger\Delta A_2 = 0$. Then we obtain

$$g'(A,b).(\Delta A,\Delta b) = L^T(A^TA)^{-1}\Delta A_2^T r - L^TA^\dagger\Delta A_1 x + L^TA^\dagger\Delta b. \tag{2.3}$$

[illegible] 2. We now prove that $\kappa_{g,F}(A,b) \leq \left\|SV^TL\right\|_2$. Let $u_i$ and $v_i$ be the $i$th column of $U$ and $V$, respectively.

From $A^\dagger = V\Sigma^{-1}U^T$, we get $AA^\dagger = UU^T = \sum_{i=1}^n u_iu_i^T$ and since $\sum_{i=1}^n v_iv_i^T = I$, we have $\Delta A_1 = \sum_{i=1}^n u_iu_i^T\Delta A$ and $\Delta A_2 = (I - AA^\dagger)\Delta A\sum_{i=1}^n v_iv_i^T$. Moreover, still using the thin SVD of $A$ and $A^\dagger$, it follows that

$$(A^TA)^{-1}v_i = \frac{v_i}{\sigma_i^2}, \quad A^\dagger u_i = \frac{v_i}{\sigma_i}, \quad \text{and} \quad A^\dagger\Delta b = \sum_{i=1}^n v_iu_i^T\frac{\Delta b}{\sigma_i}. \tag{2.4}$$

Thus (2.3) becomes

$$\begin{aligned} g'(A,b).(\Delta A,\Delta b) &= \sum_{i=1}^n L^Tv_i\left[v_i^T\Delta A^T(I - AA^\dagger)\frac{r}{\sigma_i^2} - u_i^T\Delta A\frac{x}{\sigma_i} + u_i^T\frac{\Delta b}{\sigma_i}\right] \\ &= L^T\sum_{i=1}^n v_iy_i, \end{aligned}$$

where we set $y_i = v_i^T\Delta A^T(I - AA^\dagger)\frac{r}{\sigma_i^2} - u_i^T\Delta A\frac{x}{\sigma_i} + u_i^T\frac{\Delta b}{\sigma_i} \in \mathbb{R}$.

Thus if $Y = (y_1, y_2, \ldots, y_n)^T$, we get $\|g'(A,b).(\Delta A,\Delta b)\|_2 = \left\|L^TVY\right\|_2$ and then

$$\|g'(A,b).(\Delta A,\Delta b)\|_2 = \left\|L^TVSS^{-1}Y\right\|_2 \leq \left\|SV^TL\right\|_2\left\|S^{-1}Y\right\|_2.$$

We denote by $w_i = \frac{v_i^T \Delta A^T (I - AA^\dagger) r}{S_{ii}\sigma_i^2} - \frac{u_i^T \Delta A x}{S_{ii}\sigma_i} + \frac{u_i^T \Delta b}{S_{ii}\sigma_i}$ the $i$th component of $S^{-1}Y$. Then we have

$$\begin{aligned}
|w_i| &\leq \alpha \left\| v_i^T \Delta A^T (I - AA^\dagger)^T \right\|_2 \frac{\|r\|_2}{\alpha S_{ii}\sigma_i^2} + \alpha \left\| u_i^T \Delta A \right\|_2 \frac{\|x\|_2}{\alpha S_{ii}\sigma_i} + \beta |u_i^T \Delta b| \frac{1}{\beta S_{ii}\sigma_i} \\
&\leq \left( \frac{\|r\|_2^2}{\alpha^2 S_{ii}^2 \sigma_i^4} + \frac{\|x\|_2^2}{\alpha^2 S_{ii}^2 \sigma_i^2} + \frac{1}{\beta^2 S_{ii}^2 \sigma_i^2} \right)^{\frac{1}{2}} \\
&\quad \times (\alpha^2 \left\| (I - AA^\dagger) \Delta A v_i \right\|_2^2 + \alpha^2 \left\| u_i^T \Delta A \right\|_2^2 + \beta^2 |u_i^T \Delta b|^2)^{\frac{1}{2}} \\
&= \frac{S_{ii}}{S_{ii}} (\alpha^2 \left\| (I - AA^\dagger) \Delta A v_i \right\|_2^2 + \alpha^2 \left\| u_i^T \Delta A \right\|_2^2 + \beta^2 |u_i^T \Delta b|^2)^{\frac{1}{2}}.
\end{aligned}$$

Hence

$$\begin{aligned}
\left\| S^{-1} Y \right\|_2^2 &\leq \sum_{i=1}^{n} \alpha^2 \left\| (I - AA^\dagger) \Delta A v_i \right\|_2^2 + \alpha^2 \left\| u_i^T \Delta A \right\|_2^2 + \beta^2 |u_i^T \Delta b|^2 \\
&= \alpha^2 \left\| (I - AA^\dagger) \Delta A V \right\|_F^2 + \alpha^2 \left\| U^T \Delta A \right\|_F^2 + \beta^2 \left\| U^T \Delta b \right\|_2^2 \\
&= \alpha^2 \left\| (I - AA^\dagger) \Delta A \right\|_F^2 + \alpha^2 \left\| U^T \Delta A \right\|_F^2 + \beta^2 \left\| U^T \Delta b \right\|_2^2 .
\end{aligned}$$

Since $\left\| U^T \Delta A \right\|_F = \left\| U U^T \Delta A \right\|_F = \left\| AA^\dagger \Delta A \right\|_F$ and $\left\| U^T \Delta b \right\|_2 = \left\| U U^T \Delta b \right\|_2 \leq \|\Delta b\|_2$, we get

$$\left\| S^{-1} Y \right\|_2^2 \leq \alpha^2 \left\| \Delta A_1 \right\|_F^2 + \alpha^2 \left\| \Delta A_2 \right\|_F^2 + \beta^2 \left\| \Delta b \right\|_2^2 .$$

From $\|\Delta A\|_F^2 = \|\Delta A_1\|_F^2 + \|\Delta A_2\|_F^2$, we get $\left\| S^{-1} Y \right\|_2^2 \leq \|(\Delta A, \Delta b)\|_F^2$ and thus

$$\|g'(A,b).(\Delta A, \Delta b)\|_2 \leq \left\| S V^T L \right\|_2 \|(\Delta A, \Delta b)\|_F .$$

So we have shown that $\left\| S V^T L \right\|_2$ is an upper bound for $\kappa_{g,F}(A,b)$.

*Part* 3. We now prove that this upper bound can be reached, i.e., that $\left\| S V^T L \right\|_2 = \frac{\left\| g'(A,b).(\Delta A, \Delta b) \right\|_2}{\|(\Delta A, \Delta b)\|_F}$ holds for some $(\Delta A, \Delta b) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m$.

Let us consider the particular choice of $(\Delta A, \Delta b)$ defined by

$$(\Delta A, \Delta b) = (\Delta A_2 + \Delta A_1, \Delta b) = \left( \sum_{i=1}^{n} \frac{\alpha_i}{\alpha} \frac{r}{\|r\|_2} v_i^T + \sum_{i=1}^{n} \frac{\beta_i}{\alpha} u_i \frac{x^T}{\|x\|_2}, \sum_{i=1}^{n} \frac{\gamma_i}{\beta} u_i \right),$$

where $\alpha_i$, $\beta_i$, $\gamma_i$ are real constants to be chosen in order to achieve the upper bound obtained in Part 2.

Since $\Delta A_1^T r = 0$ and $A^\dagger \Delta A_2 = 0$, it follows from (2.3) and (2.4) that

$$\begin{aligned} g'(A,b).(\Delta A, \Delta b) &= L^T (A^T A)^{-1} \sum_{i=1}^n \frac{\alpha_i}{\alpha} \|r\|_2 v_i^T - L^T A^\dagger \sum_{i=1}^n \frac{\beta_i}{\alpha} u_i \|x\|_2 \\ &\quad + L^T A^\dagger \sum_{i=1}^n \frac{\gamma_i}{\beta} u_i \\ &= L^T \sum_{i=1}^n \frac{\alpha_i}{\alpha \sigma_i^2} v_i \|r\|_2 - L^T \sum_{i=1}^n \frac{\beta_i}{\alpha \sigma_i} v_i \|x\|_2 + L^T \sum_{i=1}^n \frac{\gamma_i}{\beta \sigma_i} v_i \\ &= \sum_{i=1}^n L^T v_i \left( \frac{\alpha_i}{\alpha \sigma_i^2} \|r\|_2 - \frac{\beta_i}{\alpha \sigma_i} \|x\|_2 + \frac{\gamma_i}{\beta \sigma_i} \right). \end{aligned}$$

Thus by denoting $\xi_i = [L^T v_i \frac{\|r\|_2}{\alpha \sigma_i^2}, -L^T v_i \frac{\|x\|_2}{\alpha \sigma_i}, \frac{L^T v_i}{\beta \sigma_i}] \in \mathbb{R}^{k \times 3}$, $\Gamma = [\xi_1, \ldots, \xi_n] \in \mathbb{R}^{k \times 3n}$, and $X = (\alpha_1, \beta_1, \gamma_1, \ldots, \alpha_n, \beta_n, \gamma_n)^T \in \mathbb{R}^{3n \times 1}$ we get

$$g'(A,b).(\Delta A, \Delta b) = \Gamma X. \tag{2.5}$$

Since $\forall i, j$ $\mathrm{trace}((\frac{r}{\|r\|_2} v_i^T)^T (\frac{r}{\|r\|_2} v_i^T)) = \mathrm{trace}((u_i \frac{x^T}{\|x\|_2})^T (u_i \frac{x^T}{\|x\|_2})) = \delta_{ij}$, where $\delta_{ij}$ is the Kronecker symbol and $\mathrm{trace}((\frac{r}{\|r\|_2} v_i^T)^T (u_i \frac{x^T}{\|x\|_2})) = 0$, then $\{\frac{r}{\|r\|_2} v_i^T\}_{i=1,\ldots,n}$ and $\{u_i \frac{x^T}{\|x\|_2}\}_{i=1,\ldots,n}$ form an orthonormal set of matrices for the Frobenius norm and we get $\|\Delta A\|_F = \sum_{i=1}^n (\alpha_i^2 + \beta_i^2)$. It follows that

$$\|(\Delta A, \Delta b)\|_F^2 = \sum_{i=1}^n \alpha_i^2 + \sum_{i=1}^n \beta_i^2 + \sum_{i=1}^n \gamma_i^2 = \|X\|_2^2,$$

and (2.5) yields

$$\frac{\|g'(A,b).(\Delta A, \Delta b)\|_2}{\|(\Delta A, \Delta b)\|_F} = \frac{\|\Gamma X\|_2}{\|X\|_2}.$$

We know that $\|\Gamma\|_2 = \max_X \frac{\|\Gamma X\|_2}{\|X\|_2}$ is reached for some $X = (\alpha_1, \beta_1, \gamma_1, \ldots, \alpha_n, \beta_n, \gamma_n)^T$. Then for the $(\Delta A, \Delta b)$ corresponding to this $X$, we have $\frac{\|g'(A,b).(\Delta A, \Delta b)\|_2}{\|(\Delta A, \Delta b)\|_F} = \|\Gamma\|_2$.

Furthermore we have

$$\begin{aligned} \Gamma \Gamma^T &= L^T v_1 \left( \frac{\|r\|_2^2}{\alpha^2 \sigma_1^4} + \frac{\|x\|_2^2}{\alpha^2 \sigma_1^2} + \frac{1}{\beta^2 \sigma_1^2} \right) v_1^T L + \cdots + L^T v_n \left( \frac{\|r\|_2^2}{\alpha^2 \sigma_n^4} + \frac{\|x\|_2^2}{\alpha^2 \sigma_n^2} + \frac{1}{\beta^2 \sigma_n^2} \right) v_n^T L \\ &= L^T v_1 S_{11}^2 v_1^T L + \cdots + L^T v_n S_{nn}^2 v_n^T L \\ &= (L^T V S)(S V^T L). \end{aligned}$$

Hence

$$\|\Gamma\|_2 = \sqrt{\|\Gamma \Gamma^T\|_2} = \left\| S V^T L \right\|_2$$

and $\alpha_1, \beta_1, \gamma_1, \ldots, \alpha_n, \beta_n, \gamma_n$ are such that $\frac{\left\|g'(A,b).(\Delta A, \Delta b)\right\|_2}{\|(\Delta A, \Delta b)\|_F} = \left\|SV^T L\right\|_2$.

Thus $\left\|SV^T L\right\|_2 \leq \kappa_{g,F}(A,b)$, which concludes the proof. ☐

1. Let $l_j$ be the $j$th column of $L$, $j = 1, \ldots, k$. From

$$SV^T L = \begin{pmatrix} S_{11}{v_1}^T \\ \vdots \\ S_{nn}{v_n}^T \end{pmatrix} (l_1, \ldots, l_k) = \begin{pmatrix} S_{11}{v_1}^T l_1 & \cdots & S_{11}{v_1}^T l_k \\ \vdots & & \vdots \\ S_{nn}{v_n}^T l_1 & \cdots & S_{nn}{v_n}^T l_k \end{pmatrix},$$

it follows that $\left\|SV^T L\right\|_2$ is large when there exist at least one large $S_{ii}$ and an $l_j$ such that ${v_i}^T l_j \neq 0$. In particular, the condition number of $L^T x(A,b)$ is large when $A$ has small singular values and $L$ has components in the corresponding right singular vectors or when $\|r\|_2$ is large.

2. In the general case where $L$ is an $n \times k$ matrix, the computation of $\kappa_{g,F}(A,b)$ via the exact formula given in Theorem 1 requires the computation of the singular values and the right singular vectors of $A$, which might be expensive in practice since it involves $2mn^2$ operations if we use an R-SVD algorithm and if $m \gg n$ (see [7, p. 254]). If the LLSP is solved using a direct method, the $R$ factor of the QR decomposition of $A$ (or equivalently, in exact arithmetic, the Cholesky factor of $A^T A$) might be available. Since the right singular vectors of $A$ are also those of $R$, the condition number can be computed in about $12n^3$ flops (using the Golub–Reinsch SVD [7, p. 254]).

Using $R$ is even more interesting when $L \in \mathbb{R}^n$, since from

$$\left\|L^T A^\dagger\right\|_2 = \left\|R^{-T} L\right\|_2 \text{ and } \left\|L^T (A^T A)^{-1}\right\|_2 = \left\|R^{-1}(R^{-T} L)\right\|_2, \tag{2.6}$$

it follows that the computation of $\kappa_{g,F}(A,b)$ can be done by solving two successive $n \times n$ triangular systems which involve about $2n^2$ flops.

### 2.1. Special cases and GSVD.

In this section, we analyze some special cases of practical relevance. Moreover, we relate the formula given in Theorem 1 for

$$\kappa_{g,F}(A,b)$$

to the generalized singular value decomposition (GSVD) (see [1, p. 157], [7, p. 466], and [15, 19]). Using the GSVD of $A$ and $L^T$, there exist $U_A \in \mathbb{R}^{m \times m}, U_L \in \mathbb{R}^{k \times k}$ orthogonal matrices and $Z \in \mathbb{R}^{n \times n}$ invertible such that

$$U_A^T A = \begin{pmatrix} D_A \\ 0 \end{pmatrix} Z \quad \text{and} \quad U_L^T L^T = \begin{pmatrix} D_L & 0 \end{pmatrix} Z$$

with

$$D_A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n), \qquad D_L = \mathrm{diag}(\beta_1, \ldots, \beta_k),$$

$$\alpha_i^2 + \beta_i^2 = 1, \quad i = 1, \ldots, k, \qquad \alpha_i = 1, \quad i = k+1, \ldots, n.$$

The diagonal matrix $S$ can be decomposed in the product of two diagonal matrices

$$S = \Sigma^{-1} D$$

with

$$D_{ii} = \sqrt{\frac{{\sigma_i}^{-2} \|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}.$$

Then, taking into account the relations

$$\left\|SV^TL\right\|_2 = \left\|L^TVS\right\|_2 = \left\|L^TV\Sigma^{-1}U^TUD\right\|_2 = \left\|L^TA^\dagger UD\right\|_2,$$

$$L^TA^\dagger = U_L \begin{pmatrix} D_L & 0 \end{pmatrix} ZZ^{-1} \begin{pmatrix} D_A^{-1} & 0 \end{pmatrix} U_A^T,$$

we can represent $\kappa_{g,F}(A,b)$ as

$$\kappa_{g,F}(A,b) = \left\|T\widetilde{H}D\right\|_2,$$

where $T \in \mathbb{R}^{k\times k}$ is a diagonal matrix with $T_{ii} = \beta_i/\alpha_i$, $i = 1,\ldots,k$, and $\widetilde{H} \in \mathbb{R}^{k\times n}$ is

$$\widetilde{H} = \begin{pmatrix} I & 0 \end{pmatrix} U_A^T U.$$

Note that $\left\|L^TA^\dagger\right\|_2 = \|T\|_2$.

We also point out that the diagonal entries of $T$ are the nonzero generalized eigenvalues of

$$\lambda A^TAz = LL^Tz.$$

There are two interesting special cases where the expression of $\kappa_{g,F}(A,b)$ is simpler.

First, when $r = 0$, i.e., the LLSP problem is consistent, we have

$$D = \sqrt{\frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}\; I$$

and

$$\kappa_{g,F}(A,b) = \left\|T\widetilde{H}\right\|_2 \sqrt{\frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}.$$

Second, if we allow only perturbations on $b$ and if we use the expression (2.2) of the derivative of $g(A,b)$, we get

$$\kappa_{g,F}(A,b) = \frac{\left\|L^TA^\dagger\right\|_2}{\beta} = \frac{\|T\|_2}{\beta}$$

(see Remark 4 in section 3).

Other relevant cases where the expression for $\kappa_{g,F}(A,b)$ has a special interest are $L = I$ and $L$ is a column vector.

In the special case where $L = I$, the formula given by Theorem 1 becomes

$$\kappa_{g,F}(A,b) = \left\|SV^TL\right\|_2 = \|S\|_2 = \max_i S_{ii} = {\sigma_n}^{-1}\sqrt{\frac{{\sigma_n}^{-2}\,\|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}.$$

Since $\left\|A^\dagger\right\|_2 = {\sigma_n}^{-1}$, we obtain that

$$\kappa_{g,F}(A,b) = \left\|A^\dagger\right\|_2 \sqrt{\frac{\|A^\dagger\|_2^2\,\|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}.$$

This corresponds to the result known from [8] and also to a generalization of the formula of the condition number in the Frobenius norm given in [6, p. 92] (where only $A$ was perturbed).

Finally, let us study the particular case where $L$ is a column vector, i.e., when $g$ is a scalar derived function.

COROLLARY 1. [illegible] $L$ [illegible] $L \in \mathbb{R}^n$ [illegible] $g(A, b) = L^T x(A, b)$ [illegible]

$$\kappa_{g,F}(A, b) = \left( \left\|L^T (A^T A)^{-1}\right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\|L^T A^\dagger\right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}.$$

[illegible] By replacing $(A^T A)^{-1} = V\Sigma^{-2}V^T$ and $A^\dagger = V\Sigma^{-1}U^T$ in the expression of $K = (\left\|L^T(A^TA)^{-1}\right\|_2^2 \|r\|_2^2 + \left\|L^T A^\dagger\right\|_2^2 (\|x\|_2^2 + 1))^{\frac{1}{2}}$ we get

$$\begin{aligned} K^2 &= \left\|L^T V\Sigma^{-2}V^T\right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\|L^T V\Sigma^{-1}U^T\right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \\ &= \left\|L^T V\Sigma^{-2}\right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\|L^T V\Sigma^{-1}\right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \\ &= \left\|\Sigma^{-2}V^T L\right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\|\Sigma^{-1}V^T L\right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right). \end{aligned}$$

By writing $z = V^T L$, where $z = (z_1, \ldots, z_n)^T \in \mathbb{R}^n$, we obtain

$$\begin{aligned} K^2 &= \sum_{i=1}^{n} \frac{z_i^2}{\sigma_i^4} \frac{\|r\|_2^2}{\alpha^2} + \sum_{i=1}^{n} \frac{z_i^2}{\sigma_i^2} \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \\ &= \sum_{i=1}^{n} \frac{z_i^2}{\sigma_i^2} \left( \frac{\sigma_i^{-2}\|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \\ &= \sum_{i=1}^{n} S_{ii}^2 z_i^2 \\ &= \left\|SV^T L\right\|_2^2, \end{aligned}$$

and Theorem 1 gives the result. ☐

## 3. Sharp estimate of the partial condition number in Frobenius and spectral norms.

In many cases, obtaining a lower and/or an upper bound of $\kappa_{g,F}(A, b)$ is satisfactory when these bounds are tight enough and significantly cheaper to compute than the exact formula. Moreover, many applications use condition numbers expressed in the spectral norm. In the following theorem, we give sharp bounds for the partial condition numbers in the Frobenius and spectral norms.

THEOREM 2. [illegible] $g(A, b) = L^T x(A, b)$ $(L \in \mathbb{R}^{n\times k})$ [illegible]

$$\frac{f(A, b)}{\sqrt{3}} \leq \kappa_{g,F}(A, b) \leq f(A, b),$$

$$\frac{f(A,b)}{\sqrt{3}} \leq \kappa_{g,2}(A,b) \leq \sqrt{2} f(A,b),$$

[illegible]

$$f(A,b) = \left( \left\| L^T(A^TA)^{-1} \right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\| L^TA^\dagger \right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}.$$

[illegible] 1. We start by establishing the lower bounds. Let $w_1$ and $w_1'$ (resp., $a_1$ and $a_1'$) be the right (resp., the left) singular vectors corresponding to the largest singular values of $L^T(A^TA)^{-1}$ and $L^TA^\dagger$, respectively. We use a particular perturbation $(\Delta A, \Delta b)$ expressed as

$$(\Delta A, \Delta b) = \left( \frac{r}{\alpha \|r\|_2} w_1^T + \epsilon w_1' \frac{x^T}{\alpha \|x\|_2}, -\epsilon \frac{w_1'}{\beta} \right),$$

where $\epsilon = \pm 1$.

By replacing this value of $(\Delta A, \Delta b)$ in (2.2) we get

$$\begin{aligned} g'(A,b).(\Delta A, \Delta b) = {} & \frac{\|r\|_2}{\alpha} L^T(A^TA)^{-1} w_1 + \frac{\epsilon}{\alpha \|x\|_2} L^T(A^TA)^{-1} x {w_1'}^T r \\ & - L^TA^\dagger r \frac{w_1^T x}{\alpha \|r\|_2} - \frac{\epsilon \|x\|_2}{\alpha} L^TA^\dagger w_1' - \frac{\epsilon}{\beta} L^TA^\dagger w_1'. \end{aligned}$$

Since $r \in \operatorname{Im}(A)^\perp$ we have $A^\dagger r = 0$. Moreover we have $w_1' \in \operatorname{Ker}(L^TA^\dagger)^\perp$ and thus $w_1' \in \operatorname{Im}(A^{\dagger T} L)$, which can be written $w_1' = A^{\dagger T} L \delta$ for some $\delta \in \mathbb{R}^k$. Then ${w_1'}^T r = \delta^T L^T A^\dagger r = 0$. It follows that

$$g'(A,b).(\Delta A, \Delta b) = \frac{\|r\|_2}{\alpha} L^T(A^TA)^{-1} w_1 - \frac{\epsilon \|x\|_2}{\alpha} L^TA^\dagger w_1' - \frac{\epsilon}{\beta} L^TA^\dagger w_1'.$$

From $L^T(A^TA)^{-1} w_1 = \left\| L^T(A^TA)^{-1} \right\|_2 a_1$ and $L^TA^\dagger w_1' = \left\| L^TA^\dagger \right\|_2 a_1'$, we obtain

$$g'(A,b).(\Delta A, \Delta b) = \left\| L^T(A^TA)^{-1} \right\|_2 \frac{\|r\|_2}{\alpha} a_1 - \epsilon \left( \frac{\|x\|_2}{\alpha} + \frac{1}{\beta} \right) \left\| L^TA^\dagger \right\|_2 a_1'.$$

Since $a_1$ and $a_1'$ are unit vectors, $\|g'(A,b).(\Delta A, \Delta b)\|_2$ can be developed as

$$\begin{aligned} \|g'(A,b).(\Delta A, \Delta b)\|_2^2 = {} & \left\| L^T(A^TA)^{-1} \right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\| L^TA^\dagger \right\|_2^2 \left( \frac{\|x\|_2}{\alpha} + \frac{1}{\beta} \right)^2 \\ & - 2\epsilon \left\| L^T(A^TA)^{-1} \right\|_2 \frac{\|r\|_2}{\alpha} \left( \frac{\|x\|_2}{\alpha} + \frac{1}{\beta} \right) \left\| L^TA^\dagger \right\|_2 \cos(a_1, a_1'). \end{aligned}$$

By choosing $\epsilon = -sign(\cos(a_1, a_1'))$ the third term of the above expression becomes positive. Furthermore we have $(\frac{\|x\|_2}{\alpha} + \frac{1}{\beta})^2 \geq \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}$. Then we obtain

$$\|g'(A,b).(\Delta A, \Delta b)\|_2 \geq \left( \left\| L^T(A^TA)^{-1} \right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\| L^TA^\dagger \right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}},$$

i.e.,

$$\|g'(A,b).(\Delta A,\Delta b)\|_2 \geq f(A,b).$$

On the other hand, we have

$$\|\Delta A\|_F^2 = \left\|\frac{r}{\alpha\,\|r\|_2} w_1^T\right\|_F^2 + \left\|w_1' \frac{x^T}{\alpha\,\|x\|_2}\right\|_F^2 + 2\ \epsilon \ \text{trace}\left(\left(\frac{r}{\alpha\,\|r\|_2} w_1^T\right)^T \left(w_1' \frac{x^T}{\alpha\,\|x\|_2}\right)\right)$$

and

$$\left\|\frac{w_1'}{\beta}\right\|_2^2 = \frac{1}{\beta^2}$$

with

$$\left\|\frac{r}{\alpha\,\|r\|_2} w_1^T\right\|_F^2 = \left\|w_1' \frac{x^T}{\alpha\,\|x\|_2}\right\|_F^2 = \frac{1}{\alpha^2}, \qquad \text{trace}\left(\left(\frac{r}{\alpha\,\|r\|_2} w_1^T\right)^T \left(w_1' \frac{x^T}{\alpha\,\|x\|_2}\right)\right) = 0.$$

Then $\|(\Delta A,\Delta b)\|_F = \sqrt{3}$ and thus we have $\frac{\|g'(A,b).(\Delta A,\Delta b)\|_2}{\|(\Delta A,\Delta b)\|_F} \geq \frac{f(A,b)}{\sqrt{3}}$ for a particular value of $(\Delta A,\Delta b)$. Furthermore, from $\|(\Delta A,\Delta b)\|_2 \leq \|(\Delta A,\Delta b)\|_F$ we get $\frac{\|g'(A,b).(\Delta A,\Delta b)\|_2}{\|(\Delta A,\Delta b)\|_2} \geq \frac{f(A,b)}{\sqrt{3}}$ (for the same particular value of $(\Delta A,\Delta b)$). Then we obtain $\kappa_{g,F}(A,b) \geq \frac{f(A,b)}{\sqrt{3}}$ and $\kappa_{g,2}(A,b) \geq \frac{f(A,b)}{\sqrt{3}}$.

[illegible] 2. Let us now establish the upper bound for $\kappa_{g,F}(A,b)$ and $\kappa_{g,2}(A,b)$.

If $\Delta A_1 = AA^\dagger\Delta A$ and $\Delta A_2 = (I - AA^\dagger)\Delta A$, then it comes from (2.3) that $\forall(\Delta A,\Delta b) \in \mathbb{R}^{m\times n} \times \mathbb{R}^m$

$$\begin{aligned}\|g'(A,b).(\Delta A,\Delta b)\|_2 &\leq \left\|L^T(A^TA)^{-1}\right\|_2 \|\Delta A_2\|_2 \|r\|_2 \\ &\qquad + \left\|L^TA^\dagger\right\|_2 \|\Delta A_1\|_2 \|x\|_2 + \left\|L^TA^\dagger\right\|_2 \|\Delta b\|_2 \\ &= YX,\end{aligned}$$

where

$$Y = \left(\frac{\|L^T(A^TA)^{-1}\|_2 \|r\|_2}{\alpha}, \frac{\|L^TA^\dagger\|_2 \|x\|_2}{\alpha}, \frac{\|L^TA^\dagger\|_2}{\beta}\right)$$

and

$$X = (\alpha\,\|\Delta A_2\|_2\,, \alpha\,\|\Delta A_1\|_2\,, \beta\,\|\Delta b\|_2)^T.$$

Hence, from the Cauchy–Schwarz inequality we get

$$\|g'(A,b).(\Delta A,\Delta b)\|_2 \leq \|Y\|_2\,\|X\|_2\,, \tag{3.1}$$

with

$$\|X\|_2^2 = \alpha^2\,\|\Delta A_1\|_2^2 + \alpha^2\,\|\Delta A_2\|_2^2 + \beta^2\,\|\Delta b\|_2^2 \leq \alpha^2\,\|\Delta A_1\|_F^2 + \alpha^2\,\|\Delta A_2\|_F^2 + \beta^2\,\|\Delta b\|_2^2$$

and

$$\|Y\|_2 = f(A,b).$$

Then, since $\|\Delta A\|_F^2 = \|\Delta A_1\|_F^2 + \|\Delta A_2\|_F^2$, we have $\|X\|_2 \leq \|(\Delta A, \Delta b)\|_F$ and (3.1) yields

$$\|g'(A,b).(\Delta A, \Delta b)\|_2 \leq \|(\Delta A, \Delta b)\|_F \, \|Y\|_2 \, ,$$

which implies that

$$\kappa_{g,F}(A,b) \leq f(A,b).$$

An upper bound of $\kappa_{g,2}(A,b)$ can be computed in a similar manner: we get from (2.2) that

$$\begin{aligned} \|g'(A,b).(\Delta A, \Delta b)\|_2 &\leq (\left\|L^T(A^TA)^{-1}\right\|_2 \|r\|_2 + \left\|L^T A^\dagger\right\|_2 \|x\|_2) \, \|\Delta A\|_2 \\ &\quad + \left\|L^T A^\dagger\right\|_2 \|\Delta b\|_2 \\ &= Y'X', \end{aligned}$$

where

$$Y' = \left( \frac{\left\|L^T(A^TA)^{-1}\right\|_2 \|r\|_2 + \left\|L^T A^\dagger\right\|_2 \|x\|_2}{\alpha}, \frac{\left\|L^T A^\dagger\right\|_2}{\beta} \right)$$

and

$$X' = (\alpha \, \|\Delta A\|_2 \, , \beta \, \|\Delta b\|_2)^T.$$

Since $\|X'\|_2 = \|(\Delta A, \Delta b)\|_2$ we have $\kappa_{g,2}(A,b) \leq \|Y'\|_2$. Then using the inequality

$$\left(\left\|L^T(A^TA)^{-1}\right\|_2 \|r\|_2 + \left\|L^T A^\dagger\right\|_2 \|x\|_2\right)^2 \leq 2\left(\left\|L^T(A^TA)^{-1}\right\|_2^2 \|r\|_2^2 + \left\|L^T A^\dagger\right\|_2^2 \|x\|_2^2\right)$$

we get $\|Y'\|_2 \leq \sqrt{2} \, \|Y\|_2$ and finally obtain $\kappa_{g,2}(A,b) \leq \sqrt{2} f(A,b)$, which concludes the proof. □

Theorem 2 shows that $f(A,b)$ can be considered as a very sharp estimate of the partial condition number expressed either in Frobenius or spectral norm. Indeed, it lies within a factor $\sqrt{3}$ of $\kappa_{g,F}(A,b)$ or $\kappa_{g,2}(A,b)$.

Another observation is that we have

$$\frac{1}{\sqrt{6}} \leq \frac{\kappa_{g,F}(A,b)}{\kappa_{g,2}(A,b)} \leq \sqrt{3}.$$

Thus even if the Frobenius and spectral norms of a given matrix can be very different (for $X \in \mathbb{R}^{m \times n}$, we have $\|X\|_2 \leq \|X\|_F \leq \sqrt{n} \, \|X\|_2$), the condition numbers expressed in both norms are of the same order. The result is that a good estimate of $\kappa_{g,F}(A,b)$ is also a good estimate of $\kappa_{g,2}(A,b)$.

Moreover (2.6) shows that if the $R$ factor of $A$ is available, $f(A,b)$ can be computed by solving two $n \times n$ triangular systems with $k$ right-hand sides and thus the computational cost is $2kn^2$.

[illegible] 3. We can check in the following example that $\kappa_{g,F}(A,b)$ is not equal to $f(A,b)$. Let us consider

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \; L = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, \text{ and } \; b = \begin{pmatrix} 2/\sqrt{2} \\ 1/\sqrt{2} \\ 1 \end{pmatrix}.$$

We have

$$x = (1/\sqrt{2}, 1/\sqrt{2})^T \quad \text{and} \quad \|x\|_2 = \|r\|_2 = 1,$$

and we get

$$\kappa_{g,F}(A, b) = \frac{\sqrt{45}}{4} \; < \; f(A, b) = \frac{\sqrt{13}}{2}.$$

4. Using the definition of the condition number and of the product norms, we can obtain tight estimates for the partial condition number for perturbations of $A$ only (resp., $b$ only) by taking $\alpha > 0$ and $\beta = +\infty$ (resp., $\beta > 0$ and $\alpha = +\infty$) in Theorem 2. In particular, when we perturb only $b$ we have, with the notation of section 2.1,

$$f(A, b) = \frac{\|L^T A^\dagger\|_2}{\beta} = \frac{\|T\|_2}{\beta} = \kappa_{g,F}(A, b).$$

Moreover, when $r = 0$ we have

$$f(A, b) = \|L^T A^\dagger\|_2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right)^{\frac{1}{2}} = \|T\|_2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right)^{\frac{1}{2}}.$$

5. In the special case where $L = I$, we have

$$f(A, b) = \left( \|(A^T A)^{-1}\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \|A^\dagger\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}.$$

Since $\|(A^T A)^{-1}\|_2 = \|A^\dagger\|_2^2$ we obtain that

$$f(A, b) = \|A^\dagger\|_2 \sqrt{\frac{\|A^\dagger\|_2^2 \|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}.$$

In that case $\kappa_{g,F}(A, b)$ is exactly equal to $f(A, b)$ due to [8].

Regarding the condition number in the spectral norm, since we have $\|(\Delta A, \Delta b)\|_2 \leq \|(\Delta A, \Delta b)\|_F$ we get $\kappa_{g,2}(A, b) \geq f(A, b)$. This lower bound is similar to that obtained in [6] (where only $A$ is perturbed). As mentioned in [6], an upper bound of $\kappa_{g,2}(A)$ is $\kappa_{g,2}^u(A) = \|A^\dagger\|_2^2 \|r\|_2 + \|A^\dagger\|_2 \|x\|_2$. If we take $\alpha = 1$ and $\beta = +\infty$, we notice that $f(A, b) \leq \kappa_{g,2}^u(A) \leq \sqrt{2} f(A, b)$, showing thus that our upper bound and $\kappa_{g,2}^u(A)$ are essentially the same.

6. Generalization to other product norms:

Other product norms may have been used for the data space $\mathbb{R}^{m \times n} \times \mathbb{R}^m$.

If we consider a norm $\nu$ on $\mathbb{R}^2$ such that $c_1 \nu(x, y) \leq \sqrt{x^2 + y^2} \leq c_2 \nu(x, y)$, then we can define a product norm $\|(A, b)\|_{F,\nu} = \nu(\alpha \|\Delta A\|_F, \beta \|\Delta b\|_2)$. For instance, in [9], $\nu$ corresponds to $\|.\|_\infty$. Note that the product norm $\|(.,.)\|_F$ used throughout this paper corresponds to $\nu = \|.\|_2$ and that with the above notation we have $\|(A, b)\|_{F,2} = \|(A, b)\|_F$. Then the following inequality holds:

$$c_1 \|(\Delta A, \Delta b)\|_{F,\nu} \leq \|(\Delta A, \Delta b)\|_F \leq c_2 \|(\Delta A, \Delta b)\|_{F,\nu}.$$

If we denote $\kappa_{g,F,\nu}(A,b) = \max_{(\Delta A,\Delta b)} \frac{\left\|g'(A,b).(\Delta A,\Delta b)\right\|_2}{\|(\Delta A,\Delta b)\|_{F,\nu}}$, we obtain

$$\frac{\kappa_{g,F,\nu}(A,b)}{c_2} \leq \kappa_{g,F}(A,b) \leq \frac{\kappa_{g,F,\nu}(A,b)}{c_1}.$$

Using the bounds for $\kappa_{g,F}$ given in Theorem 2 we can obtain tight bounds for the partial condition number expressed using the product norm based on $\nu$ and when the perturbations on matrices are measured with the Frobenius norm:

$$\frac{c_1}{\sqrt{3}} f(A,b) \leq \kappa_{g,F,\nu}(A,b) \leq c_2 f(A,b).$$

Similarly, if the perturbations on matrices are measured with the spectral norm, we get

$$\frac{c_1}{\sqrt{3}} f(A,b) \leq \kappa_{g,F,\nu}(A,b) \leq c_2 \sqrt{2} f(A,b).$$

The bounds obtained for three possible product norms ($\nu = \|.\|_\infty$, $\nu = \|.\|_2$, and $\nu = \|.\|_1$) are given in Table 3.1 when using the Frobenius norm for matrices and in Table 3.2 when using the spectral norm for matrices.

Table 3.1
*Bounds for partial condition number (Frobenius norm on matrices).*

| Product norm | $\nu$, $c_1$, $c_2$ | Lower bound (factor of $f(A,b)$) | Upper bound (factor of $f(A,b)$) |
|---|---|---|---|
| $\max\{\alpha \left\|\Delta A\right\|_F, \beta \left\|\Delta b\right\|_2\}$ | $\|.\|_\infty$, $\frac{1}{\sqrt{2}}$, 1 | $\frac{1}{\sqrt{6}}$ | 1 |
| $\sqrt{\alpha^2 \left\|\Delta A\right\|_F^2 + \beta^2 \left\|\Delta b\right\|_2^2}$ | $\|.\|_2$, 1, 1 | $\frac{1}{\sqrt{3}}$ | 1 |
| $\alpha \left\|\Delta A\right\|_F + \beta \left\|\Delta b\right\|_2$ | $\|.\|_1$, 1, $\sqrt{2}$ | $\frac{1}{\sqrt{3}}$ | $\sqrt{2}$ |

Table 3.2
*Bounds for partial condition number (spectral norm on matrices).*

| Product norm | $\nu$, $c_1$, $c_2$ | Lower bound (factor of $f(A,b)$) | Upper bound (factor of $f(A,b)$) |
|---|---|---|---|
| $\max\{\alpha \left\|\Delta A\right\|_2, \beta \left\|\Delta b\right\|_2\}$ | $\|.\|_\infty$, $\frac{1}{\sqrt{2}}$, 1 | $\frac{1}{\sqrt{6}}$ | $\sqrt{2}$ |
| $\sqrt{\alpha^2 \left\|\Delta A\right\|_2^2 + \beta^2 \left\|\Delta b\right\|_2^2}$ | $\|.\|_2$, 1, 1 | $\frac{1}{\sqrt{3}}$ | $\sqrt{2}$ |
| $\alpha \left\|\Delta A\right\|_2 + \beta \left\|\Delta b\right\|_2$ | $\|.\|_1$, 1, $\sqrt{2}$ | $\frac{1}{\sqrt{3}}$ | 2 |

## 4. Statistical estimation of the partial condition number.

In this section we compute a statistical estimate of the partial condition number. We have seen in section 3 that using the Frobenius or the spectral norm for the matrices gives condition numbers that are of the same order of magnitude. For the sake of simplicity, we compute here a statistical estimate of $\kappa_{g,F}(A,b)$.

Let $(z_1, z_2, \ldots, z_q)$ be an orthonormal basis for a subspace of dimension $q$ ($q \leq k$) that has been randomly and uniformly selected from the space of all $q$-dimensional subspaces of $\mathbb{R}^k$ (this can be done by choosing $q$ random vectors and then orthogonalizing). Let us denote $g_i(A,b) = (Lz_i)^T x(A,b)$.

Since $Lz_i \in \mathbb{R}^n$, the absolute condition number of $g_i$ can be computed via the exact formula given in Corollary 1, i.e.,

$$\kappa_{g_i,F}(A,b) = \left( \left\| (Lz_i)^T (A^TA)^{-1} \right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\| (Lz_i)^T A^\dagger \right\|_2^2 \left( \frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}. \tag{4.1}$$

We define the random variable $\phi(q)$ by

$$\phi(q) = \left( \frac{k}{q} \sum_{i=1}^{q} \kappa_{g_i,F}(A,b)^2 \right)^{\frac{1}{2}}.$$

Let the operator $E(.)$ denote the expected value. The following proposition shows that the root mean square of $\phi(q)$, defined by $R(\phi(q)) = \sqrt{E(\phi(q)^2)}$, can be considered as an estimate for the condition number of $g(A,b) = L^T x(A,b)$.

Proposition 1. [illegible]

$$\frac{R(\phi(q))}{\sqrt{k}} \leq \kappa_{g,F}(A,b) \leq R(\phi(q)). \tag{4.2}$$

[illegible]. Let *vec* be the operator that stacks the columns of a matrix into a long vector and let $M$ be the $k \times m(n+1)$ matrix such that $vec(g'(A,b).(\Delta A, \Delta b)) = M\binom{vec(\alpha\Delta A)}{vec(\beta\Delta b)}$. Note that $M$ depends on $A$, $b$, $L$ and not on the $z_i$.

Then we have

$$\begin{aligned}\kappa_{g,F}(A,b) &= \max_{(\Delta A,\Delta b)} \frac{\|g'(A,b).(\Delta A,\Delta b)\|_2}{\|(\Delta A,\Delta b)\|_F} = \max_{(\Delta A,\Delta b)} \frac{\|vec(g'(A,b).(\Delta A,\Delta b))\|_2}{\left\| \begin{pmatrix} vec(\alpha\Delta A) \\ vec(\beta\Delta b) \end{pmatrix} \right\|_2} \\ &= \max_{z \in \mathbb{R}^{m(n+1)}, z \neq 0} \frac{\|M\, z\|_2}{\|z\|_2} = \|M\|_2 = \left\|M^T\right\|_2.\end{aligned}$$

Let $Z = [z_1, z_2, \ldots, z_q]$ be the $k \times q$ random matrix with orthonormal columns $z_i$. From [10] it follows that $\frac{k}{q}\left\|M^T Z\right\|_F^2$ is an unbiased estimator of the Frobenius norm of the $m(n+1) \times k$ matrix $M^T$, i.e., we have $E(\frac{k}{q}\left\|M^TZ\right\|_F^2) = \left\|M^T\right\|_F^2$.

From

$$\begin{aligned}\left\|M^TZ\right\|_F^2 &= \left\|Z^TM\right\|_F^2 \\ &= \left\| \begin{pmatrix} z_1^TM \\ \vdots \\ z_q^TM \end{pmatrix} \right\|_F^2\end{aligned}$$

we get, since $z_i^TM$ is a row vector,

$$\left\|M^TZ\right\|_F^2 = \sum_{i=1}^{q} \left\|z_i^TM\right\|_2^2.$$

We notice that for every vector $u \in \mathbb{R}^k$, if we consider the function $g_u(A,b) = u^Tg(A,b)$, then we have $\left\|u^TM\right\|_F = \|g_u'(A,b)\| = \kappa_{g_u,F}(A,b)$ and therefore

$$\left\|z_i^TM\right\|_F = \kappa_{g_i,F}(A,b).$$

Eventually we obtain

$$\left\|M^T\right\|_F^2 = E\left(\frac{k}{q}\sum_{i=1}^{q}\kappa_{g_i,F}(A,b)^2\right) = E(\phi(q)^2).$$

Moreover, considering that $M^T \in \mathbb{R}^{m(n+1)\times k}$ and using the well-known inequality

$$\frac{\left\|M^T\right\|_F}{\sqrt{k}} \leq \left\|M^T\right\|_2 \leq \left\|M^T\right\|_F,$$

we get the result (4.2). Then we will consider $\phi(q)\frac{\|(A,b)\|_F}{\|L^T\tilde{x}\|_2}$ as an estimator of $\kappa_{g,F}^{(rel)}(A,b)$. □

The root mean square of $\phi(q)$ is an upper bound of $\kappa_g(A,b)$, and estimates $\kappa_{g,F}(A,b)$ within a factor $\sqrt{k}$. Proposition 1 involves the computation of the condition number of each $g_i(A,b), i = 1,\dots,q$. From Remark 2, it follows that the computational cost of each $\kappa_{g_i,F}(A,b)$ is $2n^2$ (if the $R$ factor of the QR decomposition of $A$ is available). Hence, for a given sample of vectors $z_i$, $i = 1,\dots,q$, computing $\phi(q)$ requires about $2qn^2$ flops.

However, Proposition 1 is mostly of theoretical interest, since it relies on the computation of the root mean square of a random variable, without providing a practical method to obtain it. In the next proposition, the use of the small sample estimate theory developed by Gudmundsson, Kenney, and Laub [10] gives a first answer to this question by showing that the evaluation of $\phi(q)$ using only one sample of $q$ vectors $z_1, z_2, \dots, z_q$ in the unit sphere may provide an acceptable estimate.

PROPOSITION 2. [illegible] [10, p. 781] [illegible] $\alpha > 10$

$$\Pr\left(\frac{\phi(q)}{\alpha\sqrt{k}} \leq \kappa_{g,F}(A,b) \leq \alpha\phi(q)\right) \geq 1-\alpha^{-q}.$$

[illegible] 1 [illegible] $q$ [illegible] $\alpha = 11$ [illegible] $q = 3$ [illegible] $\phi(q)$ [illegible] $\kappa_{g,F}(A,b)$ [illegible] $11\sqrt{k}$ [illegible] 99.9%

[illegible]. We define as in the proof of Proposition 1 the matrix $M$ as the matrix related to the *vec* operation representing the linear operator $g'(A,b)$. From [10, eq. (4), p. 781 and eq. (9), p. 783] we get

$$\Pr\left(\frac{\left\|M^T\right\|_F}{\alpha} \leq \phi(q) \leq \alpha\left\|M^T\right\|_F\right) \geq 1-\alpha^{-q}. \tag{4.3}$$

We have seen in the proof of Proposition 1 that $\kappa_{g,F}(A,b) = \left\|M^T\right\|_2$. Then we have

$$\kappa_{g,F}(A,b) \leq \left\|M^T\right\|_F \leq \kappa_{g,F}(A,b)\,\sqrt{k}.$$

It follows that, for the random variable $\phi(q)$, we have

$$\Pr\left(\frac{\kappa_{g,F}(A,b)}{\alpha} \leq \phi(q) \leq \alpha\kappa_{g,F}(A,b)\,\sqrt{k}\right) \geq \Pr\left(\frac{\left\|M^T\right\|_F}{\alpha} \leq \phi(q) \leq \alpha\left\|M^T\right\|_F\right).$$

Then we obtain the result from

$$\Pr\left(\frac{\kappa_{g,F}(A,b)}{\alpha} \leq \phi(q) \leq \alpha\kappa_{g,F}(A,b)\,\sqrt{k}\right) = \Pr\left(\frac{\phi(q)}{\alpha\sqrt{k}} \leq \kappa_{g,F}(A,b) \leq \alpha\phi(q)\right). \quad \square$$

We see from this proposition that it may not be necessary to estimate the root mean square of $\phi(q)$ using sophisticated algorithms. Indeed only one sample of $\phi(q)$ obtained for $q = 3$ provides an estimate of $\kappa_{g,F}(A, b)$ within a factor $\alpha\sqrt{k}$.

[illegible] 7. If $k = 1$, then $Z = 1$ and the problem is reduced to computing $\kappa_{g_1}(A, b)$. In this case, $\phi(1)$ is exactly the partial condition number of $L^T x(A, b)$.

[illegible] 8. Concerning the computation of the statistical estimate in the presence of roundoff errors, the numerical reliability of the statistical estimate relies on an accurate computation of the $\kappa_{g_i,F}(A, b)$ for a given $z_i$. Let $A$ be a $17\times13$ Vandermonde matrix, $b$ a random vector, and $L \in \mathbb{R}^n$ the right singular vector $v_n$.

Using the [illegible] software that computes in exact arithmetic, we obtained $\kappa_{g,F}^{(rel)}(A, b) \approx 5 \cdot 10^8$. If the triangular factor $R$ form $A^T A = R^T R$ is obtained by the QR decomposition of $A$, we get $\kappa_{g,F}^{(rel)}(A, b) \approx 5 \cdot 10^8$. If $R$ is computed via a classical Cholesky factorization, we get $\kappa_{g,F}(A, b)^{(rel)} \approx 10^{10}$.

Corollary 1 and Remark 2 show that the computation of $\kappa_{g,F}(A, b)^{(rel)}$ involves linear systems of the type $A^T Ax = d$, which differs from the usual normal equation for least squares in their right-hand side. Our observation that for this kind of ill-conditioned systems, a QR factorization is more accurate than a Cholesky factorization is in agreement with [5].

## 5. Numerical experiments.

All experiments were performed in MATLAB 6.5 using a machine precision of $2.22 \cdot 10^{-16}$.

### 5.1. Examples.

For the examples of section 1, we compute the partial condition number using the formula given in Theorem 1.

In the first example we have

$$A = \begin{pmatrix} 1 & 1 & \epsilon^2 \\ \epsilon & 0 & \epsilon^2 \\ 0 & \epsilon & \epsilon^2 \\ \epsilon^2 & \epsilon^2 & 2 \end{pmatrix},$$

and we assume that only $A$ is perturbed. If we consider the values for $L$ that are $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$ and $L = (0, 0, 1)^T$, then we obtain partial condition numbers $\kappa_{g,F}^{(rel)}(A)$ that are, respectively, $10^{24}$ and 1.22, as expected since there is 50% relative error on $x_1$ and $x_2$ and there is no error on $x_3$.

In the second example where $A$ is the $10 \times 4$ Vandermonde matrix defined by $A_{ij} = \frac{1}{(10+i)^{j-1}}$ and only $b$ is perturbed, the partial condition numbers $\kappa_{g,F}^{(rel)}(b)$ with respect to each component $x_1, x_2, x_3, x_4$ are, respectively, $4.5 \cdot 10^2, 2 \cdot 10^4, 3 \cdot 10^5, 1.4 \cdot 10^6$, which is consistent with the error variation given in section 1 for each component.

### 5.2. Average behavior of the statistical estimate.

We compare here the statistical estimate described in the previous section with the partial condition number obtained via the exact formula given in Theorem 1. We suppose that only $A$ is perturbed and then the partial condition number can be expressed as $\kappa_{g,F}^{(rel)}(A)$. We use the method described in [16] in order to construct test problems $[A, x, r, b] = P(m, n, n_r, l)$ with

$$A = Y \begin{pmatrix} D \\ 0 \end{pmatrix} Z^T \in \mathbb{R}^{m\times n}, \quad Y = I - 2yy^T, \; Z = I - 2zz^T,$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ are random unit vectors and where $D = n^{-l} \operatorname{diag}(n^l, (n-1)^l, \ldots, 1)$.

$x = (1, 2^2, \ldots, n^2)^T$ is given and $r = Y\binom{0}{c} \in \mathbb{R}^m$ is computed with $c \in \mathbb{R}^{m-n}$ random vector of norm $n_r$. The right-hand side is $b = Y\binom{DZx}{c}$. By construction, the condition number of $A$ and $D$ is $n^l$.

In our experiments, we consider the matrices

$$A = \begin{pmatrix} A_1 & E' \\ E & A_2 \end{pmatrix} \quad \text{and} \quad L = \begin{pmatrix} I \\ 0 \end{pmatrix},$$

where $A_1 \in \mathbb{R}^{m_1 \times n_1}$, $A_2 \in \mathbb{R}^{m_2 \times n_2}$, $L \in \mathbb{R}^{n \times n_1}$, $m_1 + m_2 = m$, $n_1 + n_2 = n$, and $E$ and $E'$ contain the same element $e_p$ which defines the coupling between $A_1$ and $A_2$. The matrices $A_1$ and $A_2$ are randomly generated using, respectively, $P(m_1, n_1, n_{r_1}, l_1)$ and $P(m_2, n_2, n_{r_2}, l_2)$.

For each sample matrix, we compute in MATLAB

1. the partial condition number $\kappa_{g,F}^{(rel)}(A)$ using the exact formula given in Theorem 1 and based on the singular value decomposition of $A$;
2. the statistical estimate $\phi(3)$ using three random orthogonal vectors and computing each $\kappa_{g_i,F}(A, b), i = 1, 3$, with the $R$ factor of the QR decomposition of $A$.

These data are then compared by computing the ratio

$$\gamma = \frac{\phi(3)}{\kappa_{g,F}^{(rel)}(A)}.$$

Table 5.1 contains the mean $\overline{\gamma}$ and the standard deviation $s$ of $\gamma$ obtained on 1000 random matrices with $m_1 = 12, n_1 = 10, m_2 = 17, n_2 = 13$ by varying the condition numbers ${n_1}^{l_1}$ and ${n_2}^{l_2}$ of, respectively, $A_1$ and $A_2$ and the coupling coefficient $e_p$. The residual norms are set to $n_{r_1} = n_{r_2} = 1$. In all cases, $\overline{\gamma}$ is close to 1 and $s$ is about 0.3. The statistical estimate $\phi(3)$ lies within a factor 1.22 of $\kappa_{g,F}^{(rel)}(A)$, which is very accurate in condition number estimation. We notice that in two cases $\phi(3)$ is lower than 1. This is possible because Proposition 1 shows that $E(\phi(3)^2)$ is an upper bound of $\kappa_{g,F}(A)^2$ but not necessarily $\phi(3)^2$.

TABLE 5.1
*Ratio between statistical and exact condition numbers of $L^T x$.*

| Condition | | $e_p = 10^{-5}$ | | $e_p = 1$ | | $e_p = 10^5$ | |
|---|---|---|---|---|---|---|---|
| $l_1$ | $l_2$ | $\overline{\gamma}$ | $s$ | $\overline{\gamma}$ | $s$ | $\overline{\gamma}$ | $s$ |
| 1 | 1 | 1.22 | $2.28 \cdot 10^{-1}$ | 1.15 | $2.99 \cdot 10^{-1}$ | 1.07 | $3.60 \cdot 10^{-1}$ |
| 1 | 8 | 1.02 | $3.19 \cdot 10^{-1}$ | 1.22 | $3.05 \cdot 10^{-1}$ | 1.21 | $3.35 \cdot 10^{-1}$ |
| 8 | 1 | $9 \cdot 10^{-1}$ | $3 \cdot 10^{-1}$ | 1.13 | $3 \cdot 10^{-1}$ | 1.06 | $3.45 \cdot 10^{-1}$ |
| 8 | 8 | $9.23 \cdot 10^{-1}$ | $2.89 \cdot 10^{-1}$ | 1.22 | $2.95 \cdot 10^{-1}$ | 1.18 | $3.33 \cdot 10^{-1}$ |

## 6. Estimates versus exact formula.

We assume that the $R$ factor of the QR decomposition of $A$ is known. We gather in Table 6.1 the results obtained in this paper in terms of accuracy and flop counts for the estimation of the partial condition number for the LLSP. Table 6.2 gives the estimates and flop counts in the particular situation where

$$m = 1500, \; n = 1000, \; k = 50,$$

$$A_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \; L_1 = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix},$$

Table 6.1
*Comparison between exact formula and estimates for $\kappa_{g,F}(A,b)$.*

| $\kappa_{g,F}(A,b)$ | Flops | Accuracy |
|---|---|---|
| Exact formula<br>$n \ll m$ | $12n^3$ | Exact |
| Sharp estimate $f(A,b)$<br>$k \ll n$ | $2kn^2$ | $\frac{f(A,b)}{\sqrt{3}} \leq \kappa_{g,F}(A,b) \leq f(A,b)$ |
| Stat. estimate $\phi(q)$<br>$q \ll k$ | $2qn^2$ | $\frac{\phi(q)}{\alpha\sqrt{k}} \leq \kappa_{g,F}(A,b) \leq \alpha\phi(q)$<br>$Pr \geq 1-\alpha^{-q}$ for $\alpha > 10$ |

Table 6.2
*Flops and accuracy: exact formula versus estimates.*

| $\kappa_{g,F}^{(rel)}(A,b)$ | $f(A,b)\frac{\Vert(A,b)\Vert_F}{\Vert L^T\tilde{x}\Vert_2}$ | $\phi(q)\frac{\Vert(A,b)\Vert_F}{\Vert L^T\tilde{x}\Vert_2}$ |
|---|---|---|
| $2.09 \cdot 10^2$<br>12 Gflops | $2.18 \cdot 10^2$<br>100 Mflops | $11.44 \cdot 10^2$<br>6 Mflops |

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & I_{n-2} \\ 0 & 0 \end{pmatrix} \text{ and } b = \frac{1}{\sqrt{2}}(2,1,\ldots,1)^T,\ L = \begin{pmatrix} L_1 & 0 \\ 0 & I_{k-2} \\ 0 & 0 \end{pmatrix}.$$

We see here that the statistical estimates may provide information on the condition number using a very small amount of floating point operations compared with the other two methods.

**7. Conclusion.** We have shown the relevance of the partial condition number for test cases from parameter estimation. This partial condition number evaluates the sensitivity of $L^T x$, where $x$ is the solution of an LLSP when $A$ and/or $b$ are perturbed. It can be computed via a closed formula, a sharp estimate, or a statistical estimate. The choice will depend on the size of the LLSP and on the needed accuracy. The closed formula requires $\mathcal{O}(n^3)$ flops and is affordable for small problems only. The sharp estimate and the statistical estimate will be preferred for larger problems especially if $k \ll n$ since their computational cost is in $\mathcal{O}(n^2)$.

## REFERENCES

[1] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[2] Y. Cao and L. Petzold, *A subspace error estimate for linear systems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 787–801.

[3] S. Chandrasekaran and I. C. F. Ipsen, *On the sensitivity of solution components in linear systems of equations*, Numer. Linear Algebra Appl., 2 (1995), pp. 271–286.

[4] L. Eldén, *Perturbation theory for the least squares problem with linear equality constraints*, SIAM J. Numer. Anal., 17 (1980), pp. 338–350.

[5] V. Frayssé, S. Gratton, and V. Toumazou, *Structured backward error and condition number for linear systems of the type $A^*Ax = b$*, BIT, 40 (2000), pp. 74–83.

[6] A. J. Geurts, *A contribution to the theory of condition*, Numer. Math., 39 (1982), pp. 85–96.

[7] G. Golub and C. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[8] S. Gratton, *On the condition number of linear least squares problems in a weighted Frobenius norm*, BIT, 36 (1996), pp. 523–530.

[9] J. F. Grcar, *Adjoint Formulas for Condition Numbers Applied to Linear and Indefinite Least Squares*, Technical report LBNL-55221, Lawrence Berkeley National Laboratory, 2005.

[10] T. Gudmundsson, C. S. Kenney, and A. J. Laub, *Small-sample statistical estimates for matrix norms*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 776–792.

[11] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[12] E. D. Kaplan, *Understanding GPS: Principles and Applications*, Artech House, Boston, MA, 1996.

[13] C. S. Kenney and A. J. Laub, *Small-sample statistical condition estimates for general matrix functions*, SIAM J. Sci. Comput., 15 (1994), pp. 36–61.

[14] C. S. Kenney, A. J. Laub, and M. S. Reese, *Statistical condition estimation for linear least squares*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 906–923.

[15] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[16] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[17] C. R. Rao and S. K. Mitra, *Generalized Inverse of Matrices and Its Applications*, Wiley, New York, 1971.

[18] G. W. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1991.

[19] C. F. Van Loan, *Generalizing the singular value decomposition*, SIAM J. Numer. Anal., 13 (1976), pp. 76–83.

[20] Y. Wei, H. Diao, and S. Qiao, *Condition Number for Weighted Linear Least Squares Problem and Its Condition Number*, Technical report CAS 04-02-SQ, Department of Computing and Software, McMaster University, Hamilton, ON, Canada, 2004.

[21] Y. Wei, W. Xu, S. Qiao, and H. Diao, *Componentwise Condition Numbers for Generalized Matrix Inversion and Linear Least Squares*, Technical report CAS 03-12-SQ, Department of Computing and Software, McMaster University, Hamilton, ON, Canada, 2003.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 434–454

# DYNAMICAL LOW-RANK APPROXIMATION*

OTHMAR KOCH† AND CHRISTIAN LUBICH†

**Abstract.** For the low-rank approximation of time-dependent data matrices and of solutions to matrix differential equations, an increment-based computational approach is proposed and analyzed. In this method, the derivative is projected onto the tangent space of the manifold of rank-$r$ matrices at the current approximation. With an appropriate decomposition of rank-$r$ matrices and their tangent matrices, this yields nonlinear differential equations that are well suited for numerical integration. The error analysis compares the result with the pointwise best approximation in the Frobenius norm. It is shown that the approach gives locally quasi-optimal low-rank approximations. Numerical experiments illustrate the theoretical results.

**Key words.** low-rank approximation, time-varying matrices, continuous updating, smooth decomposition, matrix differential equations

**AMS subject classifications.** 65F30, 15A23

**DOI.** 10.1137/050639703

## 1. Introduction.

Low-rank approximation of unbearably large system matrices is a basic model reduction technique in many application areas, such as image compression, linear dynamical systems, regularization methods for ill-posed problems, and latent semantic indexing in information retrieval. In the present paper, we consider the task of computing low-rank approximations to matrices $A(t) \in \mathbb{R}^{m\times n}$ depending smoothly on a real parameter, henceforth referred to as time $t$. At any time $t$, a best approximation to $A(t)$ of rank $r$ is a matrix $X(t)$ in the manifold $\mathcal{M}_r = \mathcal{M}_r^{m\times n}$ of rank-$r$ matrices that satisfies

$$X(t) \in \mathcal{M}_r \quad \text{such that} \quad \|X(t) - A(t)\| = \min! \tag{1.1}$$

This is formulated for a matrix norm, which we choose as the Frobenius norm in the following. The problem is solved by a singular value decomposition (SVD) of $A(t)$, truncating all singular values after the $r$ largest ones. When the matrix is so large that a complete SVD is not feasible, a standard approach to obtaining an approximate solution is based on the Lanczos bidiagonalization process with $A(t)$ [15].

Here, we consider instead the low-rank approximation $Y(t) \in \mathcal{M}_r$ determined from the condition that for every $t$ the derivative $\dot{Y}(t)$, which is in the tangent space $\mathcal{T}_{Y(t)}\mathcal{M}_r$, be chosen as

$$\dot{Y}(t) \in \mathcal{T}_{Y(t)}\mathcal{M}_r \quad \text{such that} \quad \|\dot{Y}(t) - \dot{A}(t)\| = \min! \tag{1.2}$$

This is complemented with an initial condition, ideally $Y(t_0) = X(t_0)$. For given $Y(t)$, the derivative $\dot{Y}(t)$ is obtained by a [illegible] projection, though onto a solution-dependent vector space. Problem (1.2) yields an initial value problem of nonlinear ordinary differential equations on $\mathcal{M}_r$, which becomes numerically efficiently accessible after choosing a suitable factorization of rank-$r$ matrices.

---

*Received by the editors September 7, 2005; accepted for publication (in revised form) by H. Park September 29, 2006; published electronically April 3, 2007. This work was supported by the Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 382.

http://www.siam.org/journals/simax/29-2/63970.html

†Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D–72076 Tübingen, Germany (othmar@othmar-koch.org, lubich@na.uni-tuebingen.de).

There are several independent reasons that make the approach to low-rank approximation via (1.2) attractive:

(a) Problem (1.2) and its solution algorithm depend on the increments $\dot{A}(t)$ instead of the complete data matrix $A(t)$. This appears to be an essential benefit in processes where $\dot{A}(t)$ is much [illegible] than $A(t)$, e.g., in series of moving images or in time-varying term-document matrices in information retrieval (updates are usually small compared to the whole encyclopedia).

(b) Solving the differential equations corresponding to (1.2) requires only multiplications of $\dot{A}(t)$ with matrices having few $(r)$ columns, but [illegible] of matrices of the size of $A$, except for the low-rank approximation to the initial data $A(t_0)$.

(c) The differential equations for $Y(t)$ yield a [illegible] low-rank approximation. This is not assured when computing a pointwise best approximation, which is not unique in general.

(d) Since the problem (1.2) of determining $\dot{Y}(t)$ for given $Y(t)$ is linear, the approach extends more easily than (1.1) to [illegible] low-rank approximation, where $\mathcal{M}_r$ is replaced by some submanifold.

(e) In contrast to (1.1), the approach (1.2) extends to the situation where $A(t)$ is not a given matrix but the unknown solution of a [illegible], a matrix differential equation $\dot{A} = F(A)$. In this case, $\dot{A}(t)$ in (1.2) is simply replaced by the approximation $F(Y(t))$, so that the defect in the differential equation is minimized:

$$\dot{Y}(t) \in \mathcal{T}_{Y(t)}\mathcal{M}_r \quad \text{such that} \quad \|\dot{Y}(t) - F(Y((t))\| = \min! \tag{1.3}$$

Some comments and references to these aspects are in order: (a) and (b) are related to updating problems for low-rank approximations [3, 17], and item (c) to smooth decompositions of matrices, in particular to smooth SVD and the corresponding differential equations [1, 4, 7, 13, 16]. Item (d) refers to structured low-rank approximation as considered in [5, 6] for time-independent matrices. Item (e) and its generalization to low-rank approximation of tensors have a surprisingly long history in quantum mechanics: in 1930, Dirac [8] proposed to approximate the solution of the time-dependent Schrödinger equation, the multivariate wave function $\psi(x_1, \ldots, x_d, t)$, by a rank-1 approximation, namely an (antisymmetrized) tensor product $\phi_1(x_1, t) \ldots \phi_d(x_d, t)$, and derived differential equations for the functions $\phi_k$ from a variational principle analogous to (1.3), which is now known as the [illegible] in the chemical physics literature; see the historical references [8, 9] and, e.g., [2, 12]. Since the 1990s, the numerical approach of approximating the wave function by linear combinations of tensor products obeying differential equations derived from the Dirac–Frenkel principle (the multiconfiguration time-dependent Hartree or MCTDH method) has been used with great success for computations in quantum molecular dynamics [2]. It was, in fact, our work on variational approximations in quantum dynamics that led us to consider the dynamical low-rank matrix approximation (1.2), which does not appear to have been used or studied previously.

In the present paper we formulate the differential equations determining the solution of (1.2) and study the approximation properties of this approach, comparing the deviation from the best approximation, $Y(t) - X(t)$, with the best-approximation error $X(t) - A(t)$.

In section 2, we describe decompositions of rank-$r$ matrices and their tangent matrices, and we derive differential equations for the factors that define the rank-$r$

approximation $Y(t)$. These differential equations are used for the numerical solution of the problem. In section 3 we illustrate the approach and the behavior of the dynamical low-rank approximation (1.2) by numerical experiments.

The analysis of the approximation properties of (1.2) turns out to be more demanding than the formal similarity of (1.1) and (1.2) would suggest. In section 4 we give a preparatory result on orthogonal projections onto tangent spaces of $\mathcal{M}_r$. The approximation properties of (1.2) are then studied in section 5 under the assumption that $A(t)$ is a perturbation to a matrix of rank $\le r$. We first give near-optimality results when the effective rank of $A(t)$ is equal to $r$ (Theorems 5.1 and 5.2), and then extend the result to the case where $r$ in (1.2) is larger than the effective rank (Theorem 5.3). A further approximation result concerns systems without gaps in the distribution of the singular values (Theorem 5.5). Before turning to these approximation results, however, it should be noted that $Y(t)$ cannot always be expected to remain close to $X(t)$. This is already seen from the example of finding a rank-1 approximation to $\operatorname{diag}(e^{-t}, e^{t})$, where starting from $t_0 < 0$ yields $X(t) = Y(t) = \operatorname{diag}(e^{-t}, 0)$ for $t < 0$, but $Y(t) = \operatorname{diag}(e^{-t}, 0)$ and $X(t) = \operatorname{diag}(0, e^{t})$ for $t > 0$. The best approximation $X(t)$ here has a discontinuity at $t = 0$, caused by a crossing of singular values of which one is inside and the other outside the approximation. Our results show, however, that $Y(t)$ yields a near-optimal approximation on intervals where a good smooth approximation exists.

In section 6 we consider the following extensions of the basic approach:

- Regularization: the inverses of ill-conditioned matrices in the differential equations are replaced by regularized inverses.
- Stabilization: the differential equations are stabilized in order to drive the dynamical approximation toward the best approximation.
- Structured low-rank approximation: as an example we consider the problem of approximation by rank-$r$ orthogonal projections.
- Matrix differential equations: we extend the method and the approximation results to the low-rank approximation (1.3) to solutions of matrix differential equations $\dot{A} = F(A)$.

The present paper deals with theoretical aspects of the dynamical low-rank approximation. Our very promising first experiences in using this technique for applications ranging from the compression of time-varying term-document matrices and of series of images to the computation of blow-up in reaction-diffusion equations are reported in [14].

Throughout the paper, $\|\cdot\|$ is the Frobenius norm,

$$\|A\| = \left( \sum_{i,j} a_{ij}^2 \right)^{1/2},$$

and $\langle \cdot, \cdot \rangle$ denotes the corresponding inner product, $\langle A, B \rangle = \operatorname{tr}(A^T B) = \sum_{i,j} a_{ij} b_{ij}$. We make frequent use of the inequality $\|AB\| \le \|A\|_2 \cdot \|B\|$ and occasionally of $\|A\|_2 \le \|A\|$, where $\|\cdot\|_2$ is the spectral norm.

## 2. Differential equations for low-rank approximation.

### 2.1. Decompositions of rank-$r$ matrices and of their tangent matrices.

Every real rank-$r$ matrix of dimension $m \times n$ can be written in the form

$$Y = USV^T, \tag{2.1}$$

where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ have orthonormal columns, i.e.,

$$U^TU = I_r, \qquad V^TV = I_r \tag{2.2}$$

(with the identity matrix $I_r$ of dimension $r$), and $S \in \mathbb{R}^{r\times r}$ is nonsingular. The SVD yields $S$ diagonal, but here we will not assume a special form of $S$. The representation (2.1) is not unique: replacing $U$ by $\widetilde{U} = UP$ and $V$ by $\widetilde{V} = VQ$ with orthogonal matrices $P, Q \in \mathbb{R}^{r\times r}$, and correspondingly $S$ by $\widetilde{S} = P^TSQ$, yields the same matrix $Y = USV^T = \widetilde{U}\widetilde{S}\widetilde{V}^T$.

As a substitute for the nonuniqueness in the decomposition (2.1), we will use a [illegible]. Let $\mathcal{V}_{m,r}$ denote the Stiefel manifold of real $m \times r$ matrices with orthonormal columns. The tangent space at $U \in \mathcal{V}_{m,r}$ is

$$\mathcal{T}_U\mathcal{V}_{m,r} = \{\delta U \in \mathbb{R}^{m\times r} :\ \delta U^TU + U^T\delta U = 0\} = \{\delta U \in \mathbb{R}^{m\times r} :\ U^T\delta U \in \mathrm{so}(r)\},$$

where so$(r)$ denotes the space of skew-symmetric real $r \times r$ matrices. Consider the extended tangent map of $(S, U, V) \mapsto Y = USV^T$,

$$\begin{aligned} \mathbb{R}^{r\times r} \times \mathcal{T}_U\mathcal{V}_{m,r} \times \mathcal{T}_V\mathcal{V}_{n,r} &\to \mathcal{T}_Y\mathcal{M}_r \times \mathrm{so}(r) \times \mathrm{so}(r), \\ (\delta S, \delta U, \delta V) &\mapsto (\delta USV^T + U\delta SV^T + US\delta V^T, U^T\delta U, V^T\delta V). \end{aligned}$$

This linear map is an isomorphism, since it is readily seen to have zero null-space, and since the dimensions of the vector spaces on both sides agree.

Hence, every tangent matrix $\delta Y \in \mathcal{T}_Y\mathcal{M}_r$ is of the form

$$\delta Y = \delta USV^T + U\delta SV^T + US\delta V^T, \tag{2.3}$$

where $\delta S \in \mathbb{R}^{r\times r}$, and $\delta U \in \mathcal{T}_U\mathcal{V}_{m,r}$ and $\delta V \in \mathcal{T}_V\mathcal{V}_{n,r}$. Moreover, $\delta S, \delta U, \delta V$ are uniquely determined by $\delta Y$ if we impose the orthogonality constraints

$$U^T\delta U = 0, \qquad V^T\delta V = 0. \tag{2.4}$$

With the identity matrices $I_m$, $I_n$ of dimensions $m$ and $n$, respectively, we define by

$$P_U = UU^T, \quad P_V = VV^T, \quad P_U^{\perp} = I_m - P_U, \quad P_V^{\perp} = I_n - P_V \tag{2.5}$$

the orthogonal projections onto the spaces spanned by the columns of $U$ and $V$, and onto their orthogonal complements, respectively. Now, (2.3) and (2.4) yield

$$\begin{aligned} \delta S &= U^T\delta YV, \\ \delta U &= P_U^{\perp}\delta YVS^{-1}, \\ \delta V &= P_V^{\perp}\delta Y^TUS^{-T}. \end{aligned} \tag{2.6}$$

Formulas (2.3) and (2.6) establish an isomorphism between the subspace

$$\{(\delta S, \delta U, \delta V) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{m\times r} \times \mathbb{R}^{n\times r} : U^T\delta U = 0,\ V^T\delta V = 0\}$$

and the tangent space $\mathcal{T}_Y\mathcal{M}_r$.

**2.2. The differential equations for the factors.** The minimization condition (1.2) on the tangent space is equivalent to an orthogonal projection: find $\dot{Y} \in \mathcal{T}_Y\mathcal{M}_r$ (we omit the argument $t$) satisfying

$$\langle \dot{Y} - \dot{A}, \delta Y \rangle = 0 \quad \text{for all} \;\; \delta Y \in \mathcal{T}_Y\mathcal{M}_r. \tag{2.7}$$

From the viewpoint of numerical analysis, this is a Galerkin condition on the tangent space $\mathcal{T}_Y\mathcal{M}_r$. With this formulation we derive differential equations for the factors in the representation (2.1).

PROPOSITION 2.1. [illegible] $Y = USV^T \in \mathcal{M}_r$ [illegible] $S \in \mathbb{R}^{r\times r}$ [illegible] $U \in \mathbb{R}^{m\times r}$ [illegible] $V \in \mathbb{R}^{n\times r}$ [illegible] (1.2) [illegible] (2.7) [illegible] $\dot{Y} = \dot{U}SV^T + U\dot{S}V^T + US\dot{V}^T$, [illegible]

$$\begin{aligned} \dot{S} &= U^T\dot{A}V, \\ \dot{U} &= P_U^\perp \dot{A}VS^{-1}, \\ \dot{V} &= P_V^\perp \dot{A}^T US^{-T}, \end{aligned} \tag{2.8}$$

[illegible] $P_U^\perp = I_m - UU^T$ [illegible] $P_V^\perp = I_n - VV^T$

[illegible]. For $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, and $B \in \mathbb{R}^{m\times n}$, we use the identity

$$\langle uv^T, B\rangle = u^T Bv.$$

In view of (2.4) we require $U^T\dot{U} = V^T\dot{V} = 0$ along the solution trajectory in order to define a unique representation of $\dot{Y}$. We first substitute $\delta Y = u_i v_j^T$, for $i,\ j = 1, \dots, r$, into (2.7), where $u_i, v_j$ denote the columns of $U,\ V$, respectively. This is of the form (2.3) with $\delta U = \delta V = 0$ and one nonzero element in $\delta S$. In this way we find $\dot{S} = U^T\dot{A}V$. Similarly, choosing $\delta Y = \sum_{j=1}^r \delta u\, s_{ij} v_j^T,\ i = 1, \dots, r$, where $\delta u \in \mathbb{R}^m$ is arbitrary with $U^T\delta u = 0$, we obtain the stated differential equation for $U$, and likewise for $\delta Y = \sum_{j=1}^r u_j s_{ji} \delta v^T$ with $V^T\delta v = 0$ the differential equation for $V$. ☐

Note that with $\Lambda = U^T\dot{A}V$, the differential equations can be rewritten as

$$\begin{aligned} \dot{S} &= \Lambda, \\ \dot{U}S &= \dot{A}V - U\Lambda, \\ \dot{V}S^T &= \dot{A}^T U - V\Lambda^T. \end{aligned} \tag{2.9}$$

The matrices $U$ and $V$ retain orthonormal columns when the initial values have this property: since $U^T\dot{U} = 0$, we have $\frac{d}{dt}U^TU = \dot{U}^TU + U^T\dot{U} = 0$, and similarly for $V$.

The differential equations (2.8) are related to differential equations for other smooth matrix decompositions, in particular the smooth SVD; see [7, 16]. Unlike the differential equations for singular values given there, no singularities appear in (2.8) at points where singular values of $Y(t)$ coalesce. Equations (2.8) are in close relationship with the MCTDH equations [2], specialized to matrices instead of multivariate functions and stripped of the Schrödinger equation context.

In the numerical integration of (2.8), the step size control should be based on the local error in the low-rank approximation $Y = USV^T$, not on the local error in its factors (this makes a marked difference when $S$ has small singular values). The orthogonality of the columns of $U$ and $V$ can be preserved in the numerical integration by the methods described, e.g., in [10, Chapter IV].

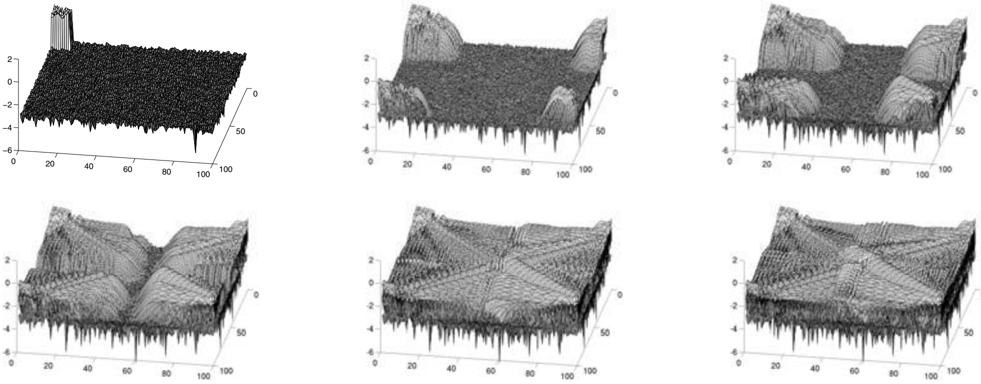


FIG. 3.1. *Size of the matrix elements for $t = 0, 0.2, \dots, 1$, first example, $\varepsilon = 1e - 3$.*

**3. Numerical experiments.** In this section we illustrate the behavior of the dynamical low-rank approximation method by three numerical examples. In all experiments, we have chosen the step sizes in the numerical integration of the differential equations (2.8) small enough that the error of the numerical integration is negligible as compared with the error of the low-rank approximation.

[illegible] We consider a model problem which was constructed in the following way: first, a $10 \times 10$ matrix of random numbers between 0 and 0.5 was added to the unit matrix of the same size, giving a matrix with singular values of magnitude $\approx 1$. Subsequently, this matrix was added as the leading $10 \times 10$ block to a $100 \times 100$ matrix with random entries between 0 and 1 multiplied by a perturbation parameter $\varepsilon$, yielding a matrix $A_1$. Another matrix $A_2$ built in the same way, multiplied by $\exp(t)$, was added for $t \in [0, 1]$. Finally, to eliminate the possibility that this particular structure of the matrix might have an influence on our results, we applied a time-dependent transformation by orthogonal matrices (which does not alter the singular values, but the left and right singular vectors), which were created by solving initial value problems $\dot{Q}_i = T_i Q_i$ $(i = 1, 2)$ with skew-symmetric $T_i$ and initial values equal to identity. To illustrate the structure of the resulting matrices,

$$A(t) = Q_1(t)(A_1 + e^t A_2)Q_2(t)^T,$$

we show, in Figure 3.1, the size of the matrix entries for values $t = 0, 0.2, 0.4, 0.6, 0.8, 1$, with the perturbation parameter $\varepsilon = 1e - 3$. At $t = 0$ the large entries are located in a corner according to the construction of the test example, and afterwards the orthogonal transformations spread these large matrix elements such that the size of matrix entries is approximately the same all over.

Figure 3.2 shows the time evolution of the errors $\|Y - X\|$, $\|Y - A\|$ and the best-approximation error $\|X - A\|$ over the interval $[0, 1]$ for $\varepsilon = 1e - 3$. When we vary the order of magnitude of the perturbation $\varepsilon$, we observe that the size of the error of the approximation $Y$ defined by (1.2) as compared with the best approximation $X$ from (1.1) is proportional to the error of the best approximation. In Table 3.1, the results are given for $\varepsilon = 1e - 1, \dots, 1e - 5$ at time $t = 1$, where the maximum errors on the interval $[0, 1]$ occur. When the parameter $\varepsilon$ is decreased by an order of magnitude, $\|X - A\|$ decreases proportionally, and $\|Y - X\|$ and $\|Y - A\|$ show the same behavior. We show the errors in the Frobenius norm, and additionally the norm of $S^{-1}$. We observe that for an approximation of rank $r = 10$, $\|S^{-1}\|$ does not increase significantly when $\varepsilon$ decreases.

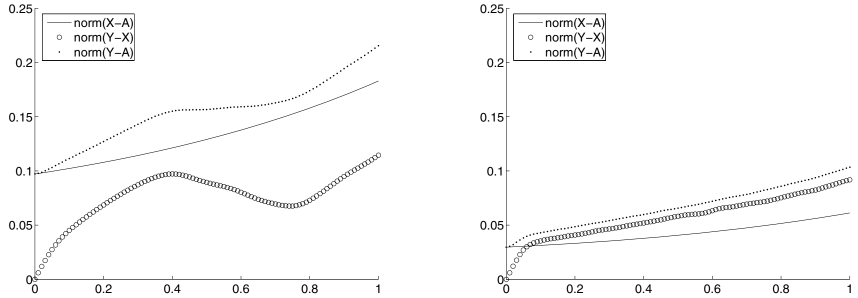


FIG. 3.2. *Errors as a function of $t$, for $r = 10$ (left) and $r = 20$ (right).*

TABLE 3.1
*First example, $r = 10$.*

| $\varepsilon$ | $\|X - A\|$ | $\|Y - X\|$ | $\|Y - A\|$ | $\|S^{-1}\|$ |
|---|---|---|---|---|
| 1e−1 | 7.3762e+00 | 1.1808e+01 | 1.3478e+01 | 7.9878e−01 |
| 1e−2 | 9.3381e−01 | 5.1817e+00 | 5.2203e+00 | 1.4487e+00 |
| 1e−3 | 1.8293e−01 | 1.1450e−01 | 2.1549e−01 | 2.6232e+00 |
| 1e−4 | 1.8310e−02 | 1.1368e−02 | 2.1550e−02 | 2.6232e+00 |
| 1e−5 | 1.8312e−03 | 1.7596e−03 | 2.5395e−03 | 2.6230e+00 |

TABLE 3.2
*First example, $r = 20$.*

| $\varepsilon$ | $\|X - A\|$ | $\|Y - X\|$ | $\|Y - A\|$ | $\|S^{-1}\|$ |
|---|---|---|---|---|
| 1e−1 | 6.0335e+00 | 1.2500e+01 | 1.3094e+01 | 1.5749e+00 |
| 1e−2 | 6.1246e−01 | 9.7993e−01 | 1.0885e+00 | 1.3569e+01 |
| 1e−3 | 6.1280e−02 | 9.1726e−02 | 1.0354e−01 | 1.3474e+02 |
| 1e−4 | 6.1282e−03 | 9.0416e−03 | 1.0298e−02 | 1.2940e+03 |

We repeat the same experiments with $r = 20$, which is larger than the effective rank 10 of the matrices that are to be approximated; see Table 3.2. We observe that $\|S^{-1}\|$ grows rapidly with $\varepsilon$ in this case. However, the approximation quality is not negatively affected. Rather, the approximation error is smaller than in the case $r = 10$, especially for larger $\varepsilon$, since more singular values are taken into account. The approximation of the dominant singular values and vectors does not suffer from the bad overall conditioning introduced by the small, insignificant contributions. For small $\varepsilon$, a similar error behavior as in the case $r = 10$ is observed. We have not included the values for still smaller $\varepsilon$, because there the numerical integrator is forced to take very small step sizes. In this way the large norm of $S^{-1}$ does have an influence on the numerical solution. This unfavorable effect can be avoided by using the regularization of section 6.

[illegible] This example demonstrates a scenario which may cause a failure of the dynamical low-rank approximation because of a discontinuous best approximation. If the approximation rank $r$ is chosen too small, then singular values which are initially small may in the course of time become larger than the singular values that are actually approximated by the algorithm. Thus, a rather large error compared to the (then discontinuous) best approximation may result when $r$ is too small and the parameter $t$ varies in an unfavorably large interval without a restart of the algorithm by recomputing a best approximation. Figure 3.3 shows such a situation for an exam-

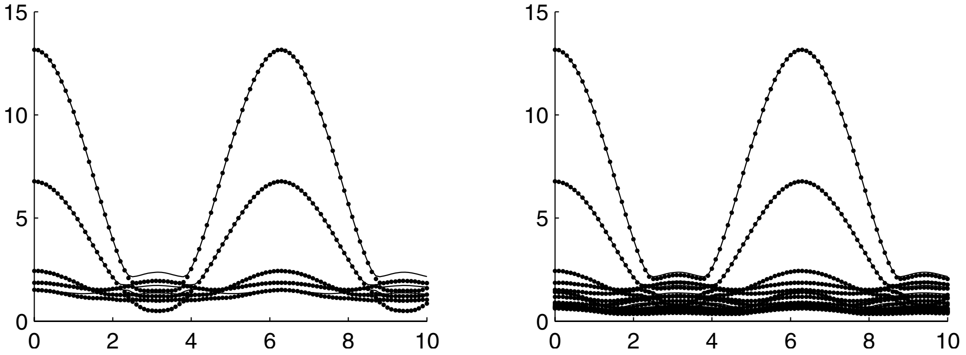


FIG. 3.3. *Second example, singular values for $r = 5$ and $r = 20$.*

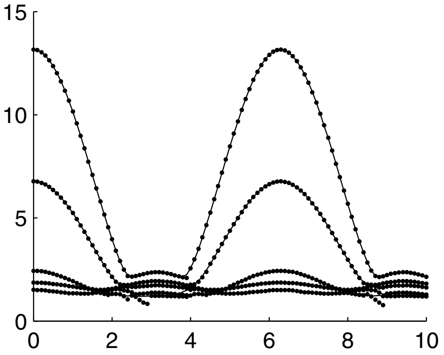


FIG. 3.4. *Second example, singular values for $r = 5$, algorithm restarted at* 20 *points.*

ple constructed by the same principles as for the example discussed above, yet with a time-dependence of the form $\cos(t)$ for $t \in [0, 10]$ and with $\varepsilon = 1e-1$. In both figures, we compare the $r$ largest singular values of $A(t)$ (computed by SVD in every point considered) with the singular values of the rank-$r$ approximation matrix $Y(t)$. In all the figures, the largest singular values computed for the exact matrix are given by a solid line, while those of the dynamical low-rank approximation (at equidistant output points) are represented by dots. If we choose $r = 5$, then the algorithm does not approximate the $r$ largest singular values for all $t$. Rather, for $t \approx 3$, one of the singular values not included in the approximation becomes largest; see Figure 3.3(left). However, if we choose $r = 20$, then all the dominant singular values and vectors are included in the approximation; see Figure 3.3(right). The correct behavior is captured with $r = 5$ if the algorithm is restarted 20 times in the interval (Figure 3.4).

[illegible] This example is again constructed similarly to the first example, $A(t) = Q_1(t)e^t DQ_2(t)^T \in \mathbb{R}^{100\times 100}$, where $D$ is a diagonal matrix with entries $2^{-i/10}$, $i = 1, \ldots, 100$, in descending order. In contrast to the first example, however, there is no distinguishable gap in the set of singular values of $A(t)$. Nonetheless, the dynamical low-rank approximation yields satisfactory results. In Figure 3.5, the errors at $t = 1$ are given for a sequence of approximations of increasing rank $r$, where $r = 3(3)99$. We observe that $\|Y - X\|$ tends to zero at the same rate as $\|X - A\|$.

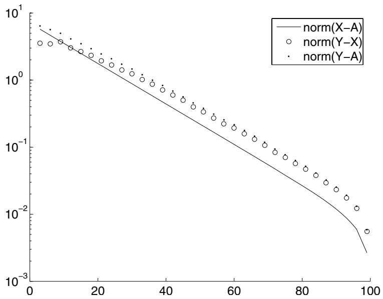


FIG. 3.5. *Third example, errors at $t = 1$ as a function of the rank $r$ of the approximation.*

**4. Tangent space projection and curvature bounds.** In the following two sections we give an analysis that explains the error behavior observed in the numerical experiments. We begin with some preparation.

Condition (2.7) can be written as the differential equation on $\mathcal{M}_r$,

$$\dot{Y} = P(Y)\dot{A}, \tag{4.1}$$

where $P(Y)$ is the orthogonal projection onto the tangent space $\mathcal{T}_Y\mathcal{M}_r$. Basic properties of this projection are formulated in the following two lemmas.

LEMMA 4.1. [illegible] $\mathcal{T}_Y\mathcal{M}_r$ [illegible] $Y = USV^T \in \mathcal{M}_r$ [illegible]

$$P(Y) = I - P^{\perp}(Y) \quad \text{[illegible]} \quad P^{\perp}(Y)B = P_U^{\perp} B P_V^{\perp} \tag{4.2}$$

[illegible] $B \in \mathbb{R}^{m\times n}$

[illegible] Proposition 2.1 yields $\dot{Y} = \dot{A} - P_U^{\perp}\dot{A}P_V^{\perp}$ for $\dot{Y}$ of (2.7) or equivalently of (4.1). Since this holds for every matrix $\dot{A}$, the result follows. □

LEMMA 4.2. [illegible] $r$ [illegible] $X \in \mathcal{M}_r$ [illegible] $\sigma_r(X) \geq \rho > 0$ [illegible] $Y \in \mathcal{M}_r$ [illegible] $\|Y - X\| \leq \frac{1}{8}\rho$ [illegible] $B \in \mathbb{R}^{m\times n}$

$$\|\big(P(Y) - P(X)\big)B\| \leq 8\rho^{-1}\,\|Y - X\| \cdot \|B\|_2, \tag{4.3}$$

$$\|P^{\perp}(Y)(Y - X)\| \leq 4\rho^{-1}\,\|Y - X\|^2. \tag{4.4}$$

[illegible] (a) For $X = U_0S_0V_0^T \in \mathcal{M}_r$ we have the bound $\|S_0^{-1}\|_2 \leq \rho^{-1}$. Since we have, by [11, p. 448],

$$|\sigma_r(Y) - \sigma_r(X)| \leq \|Y - X\|_2 \leq \|Y - X\|,$$

we obtain for $\|Y - X\| \leq \frac{1}{8}\rho$ that

$$\sigma_r(Y) \geq \sigma_r(X) - |\sigma_r(Y) - \sigma_r(X)| \geq \tfrac{7}{8}\rho,$$

and hence $Y = U_1S_1V_1^T$ with $\|S_1^{-1}\|_2 \leq \frac{8}{7}\rho^{-1}$.

(b) We decompose the matrices on the straight line connecting $X$ and $Y$ as

$$X + \tau(Y - X) = M(\tau) + N(\tau) \qquad \text{with} \qquad M(\tau) \in \mathcal{M}_r,\ N(\tau) \perp \mathcal{T}_X\mathcal{M}_r.$$

A smooth such decomposition exists at least for small $\tau$, but the arguments below show that it exists in fact for $0 \leq \tau \leq 1$. We denote

$$\Delta = P(X)(Y - X) \in \mathcal{T}_X\mathcal{M}_r, \qquad \text{with } \|\Delta\| \leq \delta := \|Y - X\|.$$

We then have $P(X)(M(\tau) - X) = \tau\Delta$, which yields

$$P(X)\dot{M}(\tau) = \Delta.$$

Since (4.2) gives $P(X)\dot{M} = \dot{M} - P_{U_0}^{\perp}\dot{M}P_{V_0}^{\perp}$, we obtain $P_{U_0}\dot{M} = P_{U_0}\Delta$ and $\dot{M}P_{V_0} = \Delta P_{V_0}$, which implies

$$U_0^T\dot{M}(\tau) = U_0^T\Delta, \quad \dot{M}(\tau)V_0 = \Delta V_0. \tag{4.5}$$

(c) Using Proposition 2.1 with $M(\tau) \in \mathcal{M}_r$ in the role of $A(\tau)$ and $Y(\tau)$, we get

$$M(\tau) = U(\tau)S(\tau)V(\tau)^T,$$

where $S, U, V$ satisfy the differential equations

$$\begin{aligned} \dot{S} &= U^T\dot{M}V = U^T\Delta V + (U-U_0)^T P_{U_0}^{\perp}\dot{M}P_{V_0}^{\perp}(V-V_0), \\ \dot{U} &= P_U^{\perp}\dot{M}VS^{-1} = P_U^{\perp}\Delta V_0 S^{-1} + P_U^{\perp}\dot{M}(V-V_0)S^{-1}, \\ \dot{V} &= P_V^{\perp}\dot{M}^T U S^{-T} = P_V^{\perp}\Delta^T U_0 S^{-T} + P_V^{\perp}\dot{M}^T(U-U_0)S^{-T}. \end{aligned} \tag{4.6}$$

In the second equalities we have used $\dot{M} = \Delta + P_{U_0}^{\perp}\dot{M}P_{V_0}^{\perp}$ and (4.5), and the fact that $P_{U_0}^{\perp}U_0 = 0$ and $P_{V_0}^{\perp}V_0 = 0$. In addition, we have $\dot{M} = U\dot{S}V^T + (\dot{U}S)V^T + U(\dot{V}S^T)^T$, and hence

$$\dot{M} = \Delta + P_{U_0}^{\perp}\big((U-U_0)\dot{S}(V-V_0)^T + (\dot{U}S)(V-V_0)^T + (U-U_0)(\dot{V}S^T)^T\big)P_{V_0}^{\perp}. \tag{4.7}$$

We will show that these differential equations have a solution up to $\tau = 1$. As long as $\|U - U_0\| \le \frac{1}{4}$ and $\|V - V_0\| \le \frac{1}{4}$, they give the bounds

$$\|\dot{S}\| \le \delta + \tfrac{1}{16}\|\dot{M}\|, \quad \|\dot{U}S\| \le \delta + \tfrac{1}{4}\|\dot{M}\|, \quad \|\dot{V}S^T\| \le \delta + \tfrac{1}{4}\|\dot{M}\|,$$

which inserted into the equation for $\dot{M}$ yield

$$\|\dot{M}\| \le 2\delta \quad \text{and} \quad \|\dot{S}\| \le \tfrac{9}{8}\delta, \quad \|\dot{U}S\| \le \tfrac{3}{2}\delta, \quad \|\dot{V}S^T\| \le \tfrac{3}{2}\delta. \tag{4.8}$$

The bound for $\dot{S}$ yields $\|S(\tau) - S_0\| \le \frac{9}{8}\delta$ for $\tau \le 1$. By the assumption $\delta \le \frac{1}{8}\rho$ and the argument in part (a), we thus obtain $\|S(\tau)^{-1}\|_2 \le \frac{4}{3}\rho^{-1}$ for $\tau \le 1$. From the bound for $\dot{U}S$ we then have

$$\|\dot{U}\| \le \|\dot{U}S\| \cdot \|S^{-1}\|_2 \le \tfrac{3}{2}\delta \cdot \tfrac{4}{3}\rho^{-1} = 2\rho^{-1}\delta \le \tfrac{1}{4}. \tag{4.9}$$

The same bound holds for $\dot{V}$. These bounds show that the differential equation has a solution on the whole interval $0 \le \tau \le 1$ with

$$\|S_1 - S_0\| \le \tfrac{9}{8}\,\delta, \quad \|U_1 - U_0\| \le 2\rho^{-1}\delta, \quad \|V_1 - V_0\| \le 2\rho^{-1}\delta. \tag{4.10}$$

(d) The above bounds give immediately

$$\|P_{U_1}^{\perp} - P_{U_0}^{\perp}\| \le 4\rho^{-1}\delta, \quad \|P_{V_1}^{\perp} - P_{V_0}^{\perp}\| \le 4\rho^{-1}\delta. \tag{4.11}$$

Formula (4.2) shows

$$\big(P(Y) - P(X)\big)B = P_{U_0}^{\perp}BP_{V_0}^{\perp} - P_{U_1}^{\perp}BP_{V_1}^{\perp},$$

which together with (4.11) yields the bound (4.3).

(e) With $P_U^\perp U = 0$ and $P_V^\perp V = 0$ we obtain

$$\begin{aligned} P^\perp(Y)(Y-X) &= P_{U_1}^\perp (U_1 S_1 V_1^T - U_0 S_0 V_0^T) P_{V_1}^\perp = -P_{U_1}^\perp U_0 S_0 V_0^T P_{V_1}^\perp \\ &= -P_{U_1}^\perp (U_1 - U_0) S_0 (V_1 - V_0)^T P_{V_1}^\perp . \end{aligned} \tag{4.12}$$

We write

$$(U_1 - U_0) S_0 = \int_0^1 \dot U(\tau) S_0 \, d\tau = \int_0^1 \dot U(\tau) S(\tau)\, d\tau - \int_0^1 \dot U(\tau)(S(\tau) - S_0)\, d\tau,$$

and hence (4.8) and (4.9) yield

$$\|(U_1 - U_0) S_0\| \le 2\delta.$$

Using this bound and (4.10) in (4.12) finally gives the bound for $P^\perp(Y)(Y-X)$. ☐

## 5. Approximation properties.

We give four results that illustrate different aspects of the dynamical low-rank approximation problem.

### 5.1. Local quasi optimality.

If the low-rank approximation problem (1.1) has a continuously differentiable best approximation $X(t) \in \mathcal{M}_r$, then the error of (1.2) can be bounded in terms of the best-approximation error $\|X(t) - A(t)\|$. The result involves a bound on $\dot A(t)$:

$$\|\dot A(t)\|_2 \le \mu \qquad \text{for } \ 0 \le t \le \bar t. \tag{5.1}$$

(For convenience we choose the initial time $t_0 = 0$.)

THEOREM 5.1. [illegible] $X(t) \in \mathcal{M}_r$ [illegible] $A(t)$ [illegible] $0 \le t \le \bar t$ [illegible] $r$ [illegible] $X(t)$ [illegible] $\sigma_r(X(t)) \ge \rho > 0$ [illegible] $\|X(t) - A(t)\| \le \frac{1}{16}\rho$ [illegible] $0 \le t \le \bar t$ [illegible] (1.2) [illegible] $Y(0) = X(0)$ [illegible]

$$\|Y(t) - X(t)\| \le 2\beta\, e^{\beta t} \int_0^t \|X(s) - A(s)\|\, ds \qquad [illegible] \quad \beta = 8\mu\rho^{-1}$$

[illegible] $t \le \bar t$ [illegible] $\frac{1}{8}\rho$

[illegible]. For the best approximation it must hold that $X - A$ is orthogonal to the tangent space $\mathcal{T}_X\mathcal{M}_r$, or equivalently,

$$P(X)(X - A) = 0.$$

We differentiate this relation with respect to $t$ and denote $(P'(X)\cdot B)\dot X = \frac{d}{dt} P(X(t))B$ to obtain

$$P(X)(\dot X - \dot A) + \big(P'(X)\cdot (X - A)\big)\dot X = 0.$$

Since $\dot X \in \mathcal{T}_X\mathcal{M}_r$, we have $P(X)\dot X = \dot X$, and the equation becomes

$$\big(I - P'(X)\cdot (X - A)\big)\dot X = P(X)\dot A. \tag{5.2}$$

Lemma 4.2 and the condition $d := \|X - A\| \le \frac{1}{16}\rho$ yield

$$\|P'(X)\cdot (X - A)\| \le 8\rho^{-1} d \le \tfrac{1}{2},$$

and hence (5.2) can be solved for $\dot{X}$ to yield

$$\dot{X} = P(X)\dot{A} + D \qquad \text{with} \quad \|D\| \le 16\rho^{-1} d\mu = 2\beta d.$$

We subtract this equation from (4.1), that is, $\dot{Y} = P(Y)\dot{A}$, and integrate from 0 to $t$. As long as $e := \|Y - X\| \le \frac{1}{8}\rho$, Lemma 4.2 yields

$$\|\big(P(Y) - P(X)\big)\dot{A}\| \le 8\rho^{-1} e\mu = \beta e,$$

and hence we obtain

$$e(t) \le \beta \int_0^t e(s)\, ds + 2\beta \int_0^t d(s)\, ds.$$

The result now follows with the Gronwall inequality. □

**5.2. A farther-reaching error bound.** Smaller errors over longer time intervals are obtained if not only $X - A$ but also its derivative is small. We assume that $A(t)$ is of the form

$$A(t) = X(t) + E(t), \qquad 0 \le t \le \bar{t}, \tag{5.3}$$

where $X(t) \in \mathcal{M}_r$ (now this need not necessarily be the best approximation) with

$$\|\dot{X}(t)\|_2 \le \mu, \tag{5.4}$$

and the derivative of the remainder term is bounded by

$$\|\dot{E}(t)\| \le \varepsilon \tag{5.5}$$

with a small $\varepsilon > 0$. We assume $\varepsilon \le \frac{1}{8}\mu$.

Theorem 5.2. [illegible] $r$ [illegible] $X(t)$ [illegible] $\sigma_r(X(t)) \ge \rho > 0$ [illegible] (1.2) [illegible] $Y(0) = X(0)$ [illegible]

$$\|Y(t) - X(t)\| \le 2t\varepsilon \qquad \text{for} \quad t \le \min\left(\bar{t}, \frac{\rho}{4\sqrt{2\mu\varepsilon}}\right).$$

[illegible]. We note $\dot{X} = P(X)\dot{X}$, rewrite (4.1) as $\dot{Y} = P(Y)\dot{X} + P(Y)\dot{E}$, and subtract the two equations. We observe

$$\big(P(Y) - P(X)\big)\dot{X} = -\big(P^\perp(Y) - P^\perp(X)\big)\dot{X} = -P^\perp(Y)\dot{X} = -P^\perp(Y)^2\dot{X}.$$

We take the inner product with $Y - X$ to obtain

$$\begin{aligned} \langle Y - X, \big(P(Y) - P(X)\big)\dot{X}\rangle &= -\langle Y - X, P^\perp(Y)\dot{X}\rangle = -\langle P^\perp(Y)(Y - X), P^\perp(Y)\dot{X}\rangle \\ &= \langle P^\perp(Y)(Y - X), \big(P(Y) - P(X)\big)\dot{X}\rangle. \end{aligned}$$

With Lemma 4.2 and (5.4), (5.5) this yields

$$\begin{aligned} \langle Y - X, \dot{Y} - \dot{X}\rangle &= \langle P^\perp(Y)(Y - X), \big(P(Y) - P(X)\big)\dot{X}\rangle + \langle Y - X, P(Y)\dot{E}\rangle \\ &\le 32\,\mu\rho^{-2}\|Y - X\|^3 + \|Y - X\| \cdot \varepsilon, \end{aligned}$$

and, on the other hand, we have

$$\langle Y-X,\dot Y-\dot X\rangle=\frac{1}{2}\frac{d}{dt}\|Y-X\|^2=\|Y-X\|\,\frac{d}{dt}\|Y-X\|.$$

Taken together, we obtain for $e(t)=\|Y(t)-X(t)\|$ the differential inequality

$$\dot e\le\gamma e^2+\varepsilon,\qquad e(0)=0,$$

with $\gamma=32\,\mu\rho^{-2}$. Hence, $e(t)$ is majorized by the solution of

$$\dot y=\gamma y^2+\varepsilon,\qquad y(0)=0,$$

which equals $y(t)=\sqrt{\varepsilon/\gamma}\,\tan(t\sqrt{\gamma\varepsilon})$ and is bounded by $2t\varepsilon$ for $t\sqrt{\gamma\varepsilon}\le 1$. Lemma 4.2 remains applicable as long as $2t\varepsilon\le\frac{1}{8}\rho$, which is satisfied on the given interval under the assumption $\varepsilon\le\frac{1}{8}\mu$. ☐

**5.3. The case of overapproximation.** The time interval in Theorem 5.2 becomes tiny when $\rho\le\varepsilon$. In that case, the effective rank ($\varepsilon$-pseudorank) of $A(t)$ is $q<r$, but the approximation is done by a rank-$r$ matrix $Y(t)$. It is not clear a priori that $Y(t)$ preserves an effective rank $q$ over longer times. Even if it does, the matrix $S(t)$ in (2.1) is ill-conditioned, and since its inverse appears in the differential equations (2.8), one might expect a severe adverse effect on the approximation properties. Remarkably, this does not happen, as is shown by the following result.

THEOREM 5.3. [illegible] (5.3)–(5.5) [illegible] $X(t)\in\mathcal{M}_q$ [illegible] $q<r$ [illegible] $q$ [illegible] $X(t)$ [illegible] $\sigma_q(X(t))\ge\rho>0$ [illegible] $Y(0)\in\mathcal{M}_r$ [illegible] $Y(0)=X(0)+E_0$ [illegible] $\operatorname{Im}E_0\perp\operatorname{Im}X(0)$ [illegible] $\operatorname{Im}E_0^T\perp\operatorname{Im}X(0)^T$ [illegible] $\|E_0\|\le\varepsilon_0\le\frac{1}{16}\rho\mu^{-1}\varepsilon$ [illegible] (2.8) [illegible] $0\le t\le t^*$ [illegible] (1.2) [illegible]

$$\|Y(t)-X(t)\|\le\varepsilon_0+6t\varepsilon\qquad\text{for}\qquad t\le\min\left(\bar t,t^*,\frac{\rho}{16\mu}\right).$$

The existence of the solution of the differential equation (2.8) is not ensured over the whole interval, since $S(t)$ might become singular. The orthogonality condition on $E_0$ is satisfied if the best rank-$r$ approximation is taken as initial value. However, this orthogonality condition is not essential. A similar, but less clear-cut estimate holds whenever $Y(0)\in\mathcal{M}_r$ is sufficiently close to $X(0)\in\mathcal{M}_q$.

The proof of Theorem 5.3 is based on combining the previous proof with the following two-scale lemma.

LEMMA 5.4. [illegible] $Y\in\mathcal{M}_r$ [illegible]

$$Y=U\begin{pmatrix}S_1&0\\0&S_2\end{pmatrix}V^T=U_1S_1V_1^T+U_2S_2V_2^T,\tag{5.6}$$

[illegible] $U=(U_1,U_2)$ [illegible] $V=(V_1,V_2)$ [illegible] $S_1\in\mathbb{R}^{q\times q}$ [illegible] $S_2\in\mathbb{R}^{(r-q)\times(r-q)}$ [illegible]

$$\sigma_{\min}(S_1)\ge\rho,\qquad\sigma_{\max}(S_2)\le\delta\qquad[\text{illegible}]\ \rho\ge 2\delta\tag{5.7}$$

[illegible] (2.7) [illegible]

$$\dot{Y} = \sum_{i=1}^{2} \Big( \dot{U}_i S_i V_i^T + U_i \dot{S}_i V_i^T + U_i S_i \dot{V}_i^T \Big), \tag{5.8}$$

[illegible]

$$\begin{aligned} \dot{S}_1 &= U_1^T \dot{A} V_1, \qquad \dot{S}_2 = U_2^T \dot{A} V_2, \\ \dot{U}_1 S_1 &= P_{U_1}^{\perp} \dot{A} V_1 + E_U, \\ \dot{V}_1 S_1^T &= P_{V_1}^{\perp} \dot{A}^T U_1 + E_V, \end{aligned} \tag{5.9}$$

[illegible] $\|E_U\| \le 2\delta\rho^{-1}\|\dot{A}\|$ [illegible] $\|E_V\| \le 2\delta\rho^{-1}\|\dot{A}\|$

The point of the lemma is that for $\delta \ll \rho$ the equations for $\dot{S}_1, \dot{U}_1, \dot{V}_1$ are, up to the small perturbations $E_U$ and $E_V$, the same as those for solving the corresponding rank-$q$ problem for $Y_q = U_1 S_1 V_1^T \in \mathcal{M}_q$; see Proposition 2.1. Under conditions (5.3)–(5.5) with $X \in \mathcal{M}_q$, the equation for $\dot{S}_2$ has a small right-hand side as long as $Y$ is close to $A$ or $X$, and hence $S_2$ remains small. The term $U_2 S_2 V_2^T$ then gives only a small contribution to $Y$, no matter what the derivatives of $U_2$ and $V_2$ are. The equations for $\dot{U}_2$ and $\dot{V}_2$, which have not been stated explicitly, contain in fact $S_2^{-1}$, which may have an arbitrarily large norm.

[illegible] 5.4. We begin by showing that $\dot{S}_i, \dot{U}_i, \dot{V}_i$ in (5.8) are uniquely determined if, instead of (2.4), we impose the constraints (cf. [7])

$$U^T \dot{U} = H, \qquad V^T \dot{V} = K,$$

with $r \times r$ matrices of the block form

$$H = \begin{pmatrix} 0 & H_{12} \\ H_{21} & 0 \end{pmatrix}, \qquad K = \begin{pmatrix} 0 & K_{12} \\ K_{21} & 0 \end{pmatrix},$$

which are skew-symmetric: $H_{12} = -H_{21}^T$ and $K_{12} = -K_{21}^T$. As in the proof of Proposition 2.1 we then obtain, instead of (2.8), the equation

$$U^T \dot{A} V = HS + \dot{S} + SK^T, \tag{5.10}$$

which yields $\dot{S}$ as the block diagonal of $\Lambda := U^T \dot{A} V$:

$$\dot{S}_1 = U_1^T \dot{A} V_1, \qquad \dot{S}_2 = U_2^T \dot{A} V_2.$$

We multiply (5.10) with $S^{-1}$ from the right and take the symmetric part. Then $H$ drops out, and we obtain

$$SK^T S^{-1} + S^{-T} K S^T = \begin{pmatrix} 0 & B_{12} S_2^{-1} \\ S_2^{-T} B_{21} & 0 \end{pmatrix},$$

where $B_{21} = B_{12}^T = \Lambda_{12}^T + S_2^T \Lambda_{21} S_1^{-1}$ is bounded by

$$\|B_{21}\| \le (1 + \delta\rho^{-1}) \, \|\dot{A}\| \le \tfrac{3}{2} \|\dot{A}\|.$$

We multiply the (2,1) block of the above equation with $S_2^T$ from the left and with $S_1^{-T}$ from the right to obtain

$$K_{21} - (S_2^T S_2) K_{21} (S_1^T S_1)^{-1} = B_{21} S_1^{-T}.$$

By condition (5.7), this equation can be uniquely solved for $K_{21}$ by fixed-point iteration, and

$$\|K_{21}\| \le \tfrac{4}{3}\rho^{-1}\|B_{21}\| \le 2\rho^{-1}\|\dot{A}\|.$$

As in (2.8), we derive

$$\dot{U}S = \dot{A}V - U\dot{S} - USK^T.$$

For the first component of $\dot{U} = (\dot{U}_1, \dot{U}_2)$ this becomes

$$\dot{U}_1 S_1 = \dot{A}V_1 - U_1\dot{S}_1 + E_U = P_{U_1}^{\perp}\dot{A}V_1 + E_U$$

with $E_U = U_2 S_2 K_{21}$, which is bounded by $\|E_U\| \le \delta \cdot 2\rho^{-1}\|\dot{A}\|$. The equation and estimate for $\dot{V}_1$ are obtained in the same way. □

In the proof of Theorem 5.3 we will actually use a variant of the above result, which is proved in the same way: if condition (5.7) is replaced by

$$\sigma_{\min}(S_1) \ge \rho, \qquad \|S_2\| \le \delta, \tag{5.11}$$

then Lemma 5.4 holds with the modified bounds

$$\|E_U\| \le 2\delta\rho^{-1}\|\dot{A}\|_2, \qquad \|E_V\| \le 2\delta\rho^{-1}\|\dot{A}\|_2. \tag{5.12}$$

*Proof of Theorem* 5.3. We write $Y$ in the form (5.6) as

$$Y = Y_1 + Y_2 \equiv U_1 S_1 V_1^T + U_2 S_2 V_2^T.$$

We will estimate $e_1 = \|Y_1 - X\|$ and $e_2 = \|Y_2\|$.

(a) By Lemma 5.4 and (5.3), $Y_1$ satisfies the differential equation

$$\dot{Y}_1 = P_q(Y_1)\dot{X} + P_q(Y_1)\dot{E} + E_U V_1^T + U_1 E_V^T,$$

where $P_q(Y_1)$ denotes the orthogonal projection onto $\mathcal{T}_{Y_1}\mathcal{M}_q$. Comparing this equation with $\dot{X} = P_q(X)\dot{X}$ as in the proof of Theorem 5.2, we obtain

$$e_1(t) \le 2t\eta, \tag{5.13}$$

provided that, up to time $t$,

$$d := \|P_q(Y_1)\dot{E} + E_U V_1^T + U_1 E_V^T\| \le \eta.$$

By (5.5) and (5.12) together with (5.4) we have

$$d(t) \le \varepsilon + 5\rho^{-1}\mu\, e_2(t). \tag{5.14}$$

(The factor 5 instead of 4 takes into account that $\|\dot{A}\| \le \mu + \varepsilon$ may be slightly larger than $\mu$ of (5.4), and $\rho$ in (5.11) may differ slightly from $\rho$ in the formulation of the theorem. The bound holds as long as $e_1$ is sufficiently small.)

(b) We have

$$e_2(t) = \|S_2(t)\| \le \varepsilon_0 + \int_0^t \|\dot{S}_2(s)\|\, ds.$$

Now,

$$\begin{aligned}\|\dot S_2\| &= \|U_2^T \dot A V_2\| = \|P_{U_2}\dot A P_{V_2}\| \le \|P_{U_1}^{\perp}\dot A P_{V_1}^{\perp}\|\\ &= \|P_q^{\perp}(Y_1)\dot A\| \le \|P_q^{\perp}(Y_1)\dot X\| + \varepsilon = \|\big(P_q^{\perp}(Y_1) - P_q^{\perp}(X)\big)\dot X\| + \varepsilon\\ &= \|\big(P_q(Y_1) - P_q(X)\big)\dot X\| + \varepsilon \le 8\rho^{-1}\mu\, e_1 + \varepsilon,\end{aligned}$$

where we have used Lemma 4.2 and (5.4) in the last inequality. Hence,

$$e_2(t) \le \varepsilon_0 + 8\rho^{-1}\mu \int_0^t e_1(s)\, ds + t\varepsilon. \tag{5.15}$$

(c) With the bound (5.13) this inequality yields

$$e_2(t) \le \varepsilon_0 + 8\rho^{-1}\mu\, t^2\eta + t\varepsilon.$$

In view of (5.14), we thus need to choose $\eta$ and the maximum value of $t$ such that

$$\eta \ge \varepsilon + 5\rho^{-1}\mu\varepsilon_0 + 5(\rho^{-1}\mu t)\varepsilon + 40(\rho^{-1}\mu t)^2\eta.$$

For $\rho^{-1}\mu\varepsilon_0 \le \frac{1}{16}\varepsilon$ and $\rho^{-1}\mu t \le \frac{1}{16}$ this is satisfied for $\eta = 2\varepsilon$. This yields

$$e_1(t) \le 4t\varepsilon, \qquad e_2(t) \le \varepsilon_0 + 2t\varepsilon,$$

which implies the result. ☐

**5.4. Systems without gaps between the singular values.** The results of the preceding subsections give satisfactory error bounds when there is a gap in the distribution of the singular values so that essential and inessential singular values are widely separated. We now consider a situation where such a gap need not exist, as in the third numerical example. We make the assumptions of Theorem 5.2 and further that $X(t) \in \mathcal{M}_r$ with $\sigma_r(X(t)) \ge \rho > 0$ has a decomposition

$$X(t) = U_0(t)S_0(t)V_0(t)^T \qquad \text{for } \ 0 \le t \le \bar t, \tag{5.16}$$

with nonsingular $S_0(t) \in \mathbb{R}^{r\times r}$, and with $U_0(t) \in \mathbb{R}^{m\times r}$ and $V_0(t) \in \mathbb{R}^{n\times r}$ having orthogonal columns, such that the following bounds hold for $0 \le t \le \bar t$:

$$\Big\|\frac{d}{dt}S_0^{-1}(t)\Big\|_2 \le c_1\rho^{-1}, \qquad \|\dot U_0(t)\|_2 \le c_2, \quad \|\dot V_0(t)\|_2 \le c_2. \tag{5.17}$$

Under these conditions we can show an $O(\varepsilon)$ error over times $O(1)$ even with $\rho \sim \varepsilon$.

THEOREM 5.5. [illegible] 5.2 [illegible] (5.16) (5.17) [illegible] (1.2) [illegible] $Y(0) = X(0)$ [illegible]

$$\|Y(t) - X(t)\| \le 2t\varepsilon \qquad \text{for} \quad t \le \min\Big(\bar t,\ \frac{1}{16c_2^{1/2}}\Big(\frac{\rho}{\varepsilon}\Big)^{1/2},\ \frac{1}{8c_1^{1/3}}\Big(\frac{\rho}{\varepsilon}\Big)^{2/3},\ \frac{1}{16}\frac{\rho}{\varepsilon}\Big).$$

[illegible] From the proof of Theorem 5.2 we have the equation

$$\langle Y - X, \dot Y - \dot X\rangle = -\langle P^{\perp}(Y)(Y - X), P^{\perp}(Y)\dot X\rangle + \langle Y - X, P(Y)\dot E\rangle. \tag{5.18}$$

For $e = \|Y - X\| \le \frac{1}{8}\rho$, the proof of Lemma 4.2 shows that $Y$ can be decomposed as $Y = U_1S_1V_1^T$ with

$$\|(U_1 - U_0)S_0\| \le 2e, \qquad \|S_0(V_1 - V_0)^T\| \le 2e.$$

By Lemma 4.1 we can write

$$\begin{aligned} P^\perp(Y)\dot X &= P_{U_1}^\perp\big(\dot U_0 S_0 V_0^T + U_0 \dot S_0 V_0^T + U_0 S_0 \dot V_0^T\big)P_{V_1}^\perp \\ &= P_{U_1}^\perp\big(-\dot U_0 S_0 (V_1 - V_0)^T + (U_1 - U_0)S_0 \cdot S_0^{-1}\dot S_0 S_0^{-1} \cdot S_0 (V_1 - V_0)^T \\ &\qquad - (U_1 - U_0) S_0 \dot V_0^T\big)P_{V_1}^\perp . \end{aligned}$$

With (4.4), (5.17), and the above estimate, (5.18) gives the differential inequality, as long as $e \le \frac{1}{8}\rho$:

$$\dot e \le 4\rho^{-1} e\big(2c_2 e + 4c_1\rho^{-1}e^2 + 2c_2 e\big) + \varepsilon. \tag{5.19}$$

The error is bounded by $2t\varepsilon$ as long as the first term on the right-hand side is bounded by $\varepsilon$, which is thus satisfied for $16c_2\rho^{-1}(2t\varepsilon)^2 \le \frac{1}{2}\varepsilon$ and $16c_1\rho^{-2}(2t\varepsilon)^3 \le \frac{1}{2}\varepsilon$. This holds under the stated bounds for $t$. ◻

## 6. Extensions of the basic approach.

**6.1. Regularization.** Though Theorem 5.3 shows that overapproximation has no disastrous effect on the approximation properties, it is in fact harmful to the numerical solution of the differential equations (2.8). Near-singularity of $S$ enforces very small step sizes in numerical integrators, and rounding errors may become important. The situation is alleviated by replacing $S^{-1}$ by a [illegible], e.g., obtained from computing an SVD of $S \in \mathbb{R}^{r\times r}$ and replacing the $i$th singular value $\sigma_i$ by $\sqrt{\sigma_i^2 + \epsilon^2}$. The approximation result of Theorem 5.3 remains valid (with modified constants), since transformation to the block form (5.8) yields only an $O(\epsilon)$ perturbation in (5.9).

**6.2. Stabilization.** In order to drive the solution toward the best approximation, we replace (2.7) by

$$\langle \dot Y - \dot A, \delta Y\rangle + \alpha\langle Y - A, \delta Y\rangle = 0 \qquad \text{for all} \quad \delta Y \in \mathcal{T}_Y\mathcal{M}_r \tag{6.1}$$

with a positive parameter $\alpha$. This amounts to replacing $\dot A$ by $\dot A - \alpha(Y - A)$ in the differential equations (2.8) determining $Y = USV^T$. (This approach requires knowledge of both $\dot A$ and $A$ and can therefore not be extended to the low-rank approximation of matrix differential equations as in section 6.4 below.)

The effect of the parameter $\alpha$ is easily seen in the framework of the proof of Theorem 5.1. With the notation used there, we have the differential equations

$$\begin{aligned} \dot Y &= P(Y)(\dot A - \alpha(Y - A)), \\ \dot X &= P(X)(\dot A - \alpha(X - A)) + D. \end{aligned}$$

Subtracting the equations yields

$$\begin{aligned} \dot Y - \dot X = \big(P(Y) - P(X)\big)\dot A - D \\ - \alpha(Y - X) + \alpha P^\perp(Y)(Y - X) - \alpha\big(P(Y) - P(X)\big)(X - A). \end{aligned}$$

Taking the inner product with $Y - X$ and using Lemma 4.2 yields the following differential inequality for $e = \|Y - X\|$: with $d = \|X - A\|$ and $\beta = 8\mu\rho^{-1}$,

$$\dot e \le \beta e + 2\beta d - \alpha e(1 - \tfrac{1}{2}\beta e - \beta d). \tag{6.2}$$

The last term is stabilizing, provided that $d$ and $e$ are small enough.

**6.3. An example of structured low-rank approximation: Approximation on Grassmann manifolds.** We now approximate $A(t) \in \mathbb{R}^{n\times n}$ not just by arbitrary rank-$r$ matrices, but by orthogonal projections onto $r$-dimensional subspaces. We thus replace the manifold $\mathcal{M}_r^{n\times n}$ in (1.2) by the submanifold (known as a Grassmann manifold)

$$\mathcal{G} = \mathcal{G}_{n,r} = \{Y \in \mathcal{M}_r^{n\times n} :\ Y^2 = Y,\ Y^T = Y\}.$$

A projection $Y \in \mathcal{G}$ can be written, in a nonunique way, as

$$Y = UU^T \qquad \text{with} \quad U \in \mathcal{V}_{n,r};$$

that is, $U \in \mathbb{R}^{n\times r}$ has orthonormal columns. $U$ is unique up to right-multiplication with an $r \times r$ orthogonal matrix. Tangent matrices in $\mathcal{T}_Y\mathcal{G}$ are of the form

$$\delta Y = \delta U U^T + U\delta U^T \qquad \text{with} \quad \delta U \in \mathcal{T}_U\mathcal{V}_{n,r}. \tag{6.3}$$

This representation is unique if we impose the condition $U^T\delta U = 0$, which yields $\delta U^T = U^T\delta Y$. The Galerkin condition (2.7) for the manifold $\mathcal{G}$ determines $\dot{Y} \in \mathcal{T}_Y\mathcal{G}$ such that

$$\langle \dot{Y} - \dot{A}, \delta Y\rangle = 0 \qquad \text{for all} \ \ \delta Y \in \mathcal{T}_Y\mathcal{G}.$$

Substituting (6.3) and using the rules $\langle A, B\rangle = \langle A^T, B^T\rangle$ and $\langle A, BC^T\rangle = \langle AC, B\rangle$, this condition becomes, with the condition $U^T\dot{U} = 0$,

$$\langle \dot{U} - \tfrac{1}{2}(\dot{A} + \dot{A}^T)U,\ \delta U\rangle = 0 \quad \text{ for all } \delta U \in \mathbb{R}^{n\times r} \text{ with } U^T\delta U = 0.$$

This gives the differential equation

$$\dot{U} = P_U^{\perp}\tfrac{1}{2}(\dot{A} + \dot{A}^T)U. \tag{6.4}$$

With the appropriate version of Lemma 4.2 for the orthogonal projection $P(Y)$ onto the submanifold $\mathcal{G}$, the approximation estimates corresponding to Theorems 5.1 and 5.2 follow without further ado.

**6.4. Minimum defect approximation of matrix differential equations.** For the low-rank approximation to a solution of the matrix differential equation

$$\dot{A} = F(A), \tag{6.5}$$

condition (1.2) is replaced, at every time $t$, by

$$\dot{Y} \in \mathcal{T}_Y\mathcal{M}_r \quad \text{ such that } \quad \|\dot{Y} - F(Y)\| = \min! \tag{6.6}$$

Equivalently, condition (2.7) is replaced by the Galerkin condition

$$\langle \dot{Y} - F(Y),\ \delta Y\rangle = 0 \quad \text{for all} \ \ \delta Y \in \mathcal{T}_Y\mathcal{M}_r, \tag{6.7}$$

and correspondingly, the expression $\dot{A}$ is replaced by $F(Y)$ for $Y = USV^T$ in the differential equations (2.8) for $S, U, V$.

Theorems 5.1–5.3 extend to the low-rank approximation of matrix differential equations (6.5). We assume that $F$ has a moderate bound along the approximations,

$$\|F(X(t))\| \le \mu, \qquad \|F(Y(t))\| \le \mu \qquad \text{for} \ \ 0 \le t \le \bar{t}, \tag{6.8}$$

and satisfies a one-sided Lipschitz condition: there is a real $\lambda$ (positive or negative or zero) such that

$$\langle F(Y) - F(X), Y - X \rangle \le \lambda \, \|Y - X\|^2 \tag{6.9}$$

for all matrices $X, Y \in \mathcal{M}_r$. We further assume that for the best approximation $X(t)$,

$$\|F(X(t)) - F(A(t))\| \le L \, \|X(t) - A(t)\| \qquad \text{for } \ 0 \le t \le \bar{t}, \tag{6.10}$$

which is in particular satisfied if $F$ is Lipschitz continuous with Lipschitz constant $L$. We then have the following extension of the quasi-optimality result of Theorem 5.1.

THEOREM 6.1. [illegible] $X(t) \in \mathcal{M}_r$ [illegible] $A(t)$ [illegible] (6.5) [illegible] $0 \le t \le \bar{t}$ [illegible] (6.8) [illegible] (6.10) [illegible] $r$ [illegible] $X(t)$ [illegible] $\sigma_r(X(t)) \ge \rho > 0$ [illegible] $\|X(t) - A(t)\| \le \frac{1}{16}\rho$ [illegible] $0 \le t \le \bar{t}$ [illegible] (1.2) [illegible] $Y(0) = X(0)$ [illegible]

$$\|Y(t) - X(t)\| \le (2\beta + L)\, e^{(2\beta+\lambda)t} \int_0^t \|X(s) - A(s)\| \, ds \qquad \text{[illegible]} \quad \beta = 8\mu\rho^{-1}$$

[illegible] $t \le \bar{t}$ [illegible] $\frac{1}{8}\rho$

[illegible]. Equation (6.7) rewritten as in (4.1) reads

$$\dot{Y} = P(Y)F(Y). \tag{6.11}$$

As in the proof of Theorem 5.1, we have the equation

$$\dot{X} = P(X)F(A) + D \qquad \text{with} \quad \|D\| \le 2\beta d$$

for $d = \|X - A\|$. We subtract the two equations, write

$$\begin{aligned} P(Y)F(Y) - P(X)F(A) - D = {} & (P(Y) - P(X))F(X) + P(X)(F(X) - F(A)) \\ & + (F(Y) - F(X)) - P^{\perp}(Y)(F(Y) - F(X)) - D, \end{aligned}$$

and take the inner product with $Y - X$. With Lemma 4.2 we obtain

$$\begin{aligned} \langle \dot{Y} - \dot{X}, Y - X \rangle \le {} & \beta \, \|Y - X\|^2 + Ld \, \|Y - X\| \\ & + \lambda \, \|Y - X\|^2 + \beta \, \|Y - X\|^2 + 2\beta d \, \|Y - X\|. \end{aligned}$$

For $e = \|Y - X\|$ this gives the differential inequality

$$\dot{e} \le (2\beta + \lambda)e + (2\beta + L)d, \qquad e(0) = 0, \tag{6.12}$$

which yields the result. □

We refer to [12, Theorem 4.1] for a related quasi-optimality result in a situation of a linear differential equation with an unbounded operator.

In the differential equation analogue of Theorem 5.2 with the splitting (5.3), we start from the equations $\dot{Y} - \dot{X} = P(Y)F(Y) - P(X)\dot{X}$ and $\dot{X} = F(A) - \dot{E}$, yielding

$$\begin{aligned} \dot{Y} - \dot{X} = {} & (P(Y) - P(X))\dot{X} - P^{\perp}(Y)(F(Y) - F(X)) \\ & + (F(Y) - F(X)) + P(Y)(F(X) - F(A)) + P(Y)\dot{E}, \end{aligned}$$

where we now take the inner product with $Y - X$. If it is additionally assumed that $F$ has Lipschitz constant $L$, then this leads to the differential inequality

$$\dot{e} \leq 4\rho^{-1}(\beta + L)e^2 + \lambda e + Ld + \varepsilon, \qquad e(0) = 0. \tag{6.13}$$

With $\widehat{\gamma} = 4\rho^{-1}(\beta + L)$ and $\widehat{\varepsilon} = \varepsilon + L \max_{0 \leq t \leq \bar{t}} d(t)$, and with $\varphi(x) = (e^x - 1)/x$, this yields the error bound

$$\|Y(t) - X(t)\| \leq 2t\,\varphi(\lambda t)\,\widehat{\varepsilon} \qquad \text{for} \quad t\varphi(\lambda t) \leq \tfrac{1}{2}(\widehat{\gamma}\widehat{\varepsilon})^{-1/2} \tag{6.14}$$

and as long as $t \leq \bar{t}$ and $2t\varphi(\lambda t)\widehat{\varepsilon} \leq \frac{1}{8}\rho$.

Theorems 5.3 and 5.5 are extended similarly.

**6.5. The special case of linear matrix differential equations.** Systems

$$\dot{A} = LA + AR \tag{6.15}$$

with possibly time-dependent matrices $L(t)$ and $R(t)$ have the solution $A(t) \in \mathcal{M}_r$ for initial data $A_0 = Y_0 \in \mathcal{M}_r$. This is seen immediately from Lemma 4.1 and (6.11), which yield $\dot{Y} = LY + YR$ and hence $A(t) = Y(t) \in \mathcal{M}_r$. From the differential equations (2.8), we thus obtain a decomposition of the solution $A = USV^T$ with $U$ and $V$ having orthonormal columns and with the factors satisfying the differential equations

$$\begin{aligned} \dot{S} &= U^T LUS + SV^T RV, \\ \dot{U} &= P_U^{\perp} LU, \\ \dot{V} &= P_V^{\perp} R^T V. \end{aligned} \tag{6.16}$$

A different situation arises for linear problems of the type

$$\dot{A} = LA + AR + B \bullet A, \tag{6.17}$$

where $\bullet$ denotes the Hadamard (or entrywise) product of matrices. The differential equations (2.8), with $\dot{A}$ replaced by the right-hand side of (6.17) evaluated at $Y = USV^T$ instead of $A$, determine a low-rank approximation, but the entrywise multiplication with $B(t)$ in general requires the explicit computation of the entries of $Y$. The situation simplifies if $B$ is itself a low-rank matrix $B = \sum_{j=1}^{k} \beta_j c_j d_j^T$. Writing $Y = \sum_{i=1}^{r} \sigma_i \widehat{u}_i \widehat{v}_i^T$ (obtained from an SVD of the matrix $S$ of $Y = USV^T$), we can use $B \bullet Y$ in the differential equations for $S, U, V$ in the decomposed form

$$B \bullet Y = \sum_{j=1}^{k} \sum_{i=1}^{r} \beta_j \sigma_i (c_j \bullet \widehat{u}_i)(d_j \bullet \widehat{v}_i)^T;$$

cf. [2] for an analogous observation for the potential in the Schrödinger equation. The dynamical low-rank approximation $Y(t)$ to $A(t)$ of (6.17) can thus be computed inexpensively if $B(t)$ is of low rank or otherwise approximated by a matrix of low rank, and in the present paper we have seen how this can be computed.

**7. Conclusions and outlook.** The dynamical low-rank approximation (1.2), or equivalently (2.7) or (4.1), becomes an attractive computational approach via the differential equations (2.8) that determine the factors in the representation (2.1) of the approximation. The method yields a near-optimal smooth low-rank approximation, as is shown in Theorems 5.1, 5.2, 5.3, and 5.5 and observed in numerical experiments. A direct but very noteworthy extension is the minimum-defect low-rank approximation (1.3) to solutions of matrix differential equations. Our first numerical experience in compressing time-varying term-document matrices and series of images, and in approximating time-dependent PDEs whose solutions are essentially of low rank (e.g., smooth with the exception of a few pulses or spikes, as in blow-up problems in reaction-diffusion equations), is very promising, as reported in [14]. It will be interesting to see the dynamical low-rank approximation used for large-scale problems in applications, well beyond the already important area of quantum dynamics, where basic ideas for this approach originated 75 years ago as a physical model reduction technique.

## REFERENCES

[1] M. Baumann and U. Helmke, *Singular value decomposition of time-varying matrices*, Future Generation Computer Systems, 19 (2003), pp. 353–361.

[2] M. H. Beck, A. Jäckle, G. A. Worth, and H.-D. Meyer, *The multiconfiguration time-dependent Hartree (MCTDH) method: A highly efficient algorithm for propagating wavepackets*, Phys. Rep., 324 (2000), pp. 1–105.

[3] M. W. Berry, S. T. Dumais, and G. W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Rev., 37 (1995), pp. 573–595.

[4] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numer. Math., 60 (1991), pp. 1–39.

[5] M. T. Chu, R. E. Funderlic, and R. J. Plemmons, *Structured low rank approximation*, Linear Algebra Appl., 366 (2003), pp. 157–172.

[6] M. Chu, N. Del Buono, L. Lopez, and T. Politi, *On the low-rank approximation of data on the unit sphere*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 46–60.

[7] L. Dieci and T. Eirola, *On smooth decompositions of matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 800–819.

[8] P. A. M. Dirac, *Note on exchange phenomena in the Thomas atom*, Proc. Cambridge Phil. Soc., 26 (1930), pp. 376–385.

[9] J. Frenkel, *Wave Mechanics, Advanced General Theory*, Clarendon Press, Oxford, 1934.

[10] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration, Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed., Springer, Berlin, 2006.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, London, 1985.

[12] C. Lubich, *On variational approximations in quantum molecular dynamics*, Math. Comp., 74 (2005), pp. 765–779.

[13] V. Mehrmann and W. Rath, *Numerical methods for the computation of analytic singular value decompositions*, Electron. Trans. Numer. Anal., 1 (1993), pp. 72–88.

[14] A. Nonnenmacher and C. Lubich, *Dynamical Low-Rank Approximation: Applications and Numerical Experiments*, technical report, Mathematics Institut, University of Tübingen, 2006; available online at http://na.uni-tuebingen.de/preprints.shtml.

[15] H. D. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.

[16] K. Wright, *Differential equations for the analytic singular value decomposition of a matrix*, Numer. Math., 63 (1992), pp. 283–295.

[17] H. Zha and H. D. Simon, *On updating problems in latent semantic indexing*, SIAM J. Sci. Comput., 21 (1999), pp. 782–791.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 455–472

# HERMITE INDICES AND JORDAN STRUCTURE OF A PERTURBED LINEAR SYSTEM[*]

I. BARAGAÑA[†], V. FERNÁNDEZ[†], AND I. ZABALLA[‡]

**Abstract.** The aim of this work is to characterize the Hermite indices and the Jordan structure of a pair of matrices $(A, B)$ under small perturbations. The proofs of the sufficiency give us a constructive method to obtain a pair as close as we want to $(A, B)$ and with prescribed structure.

**Key words.** similarity, Hermite indices, invariant factors, Segre characteristic

**AMS subject classifications.** 93B05, 93B10

**DOI.** 10.1137/060650726

## 1. Introduction.

Consider the following system of differential equations with control:

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{1.1}$$

where $A \in \mathbb{F}^{n\times n}$ and $B \in \mathbb{F}^{n\times m}$, with $\mathbb{F}$ the field of real or complex numbers. We will identify the system with the matrix pair $(A, B)$.

A change of bases in the state space, $y = Px$, yields the new system $\dot{y}(t) = PAP^{-1}y(t) + PBu(t)$. Pairs $(A, B)$ and $(PAP^{-1}, PB)$ are said to be similar. For controllable systems there may be many complete systems of invariants (see [15, 19]), all of them consisting of two sequences of numbers: a sequence of real or complex numbers and a, possibly nonordered, sequence of positive integers whose sum is $n$, the order of the system (i.e., a partition of $n$). The elements of these partitions are indices of nice bases [1] taken from the controllability matrix of the system and appear as the size of prominent blocks in the corresponding canonical form of $(A, B)$. Following [11] these invariants can be called invariants of structure of the system. For noncontrollable systems there are, in addition, other invariants of structure: the partial multiplicities of the eigenvalues of $(A, B)$, i.e., the sizes of the blocks in the Jordan canonical form of the noncontrollable part of the system.

The aim of this paper is to study the change of some invariants of structure of $(A, B)$ under small additive perturbations. This can be seen as a continuation of the results in [3, 6, 12, 16]. In [3, 12] the authors solved the problem posed in [5] on the possible partial multiplicities of the eigenvalues of all square matrices that are close enough to a given square matrix over $\mathbb{C}$. Also they proved that the obtained conditions are necessary and sufficient for the existence of matrices, as close as one may desire to a given matrix. In [2] the problem of characterizing the closure of similarity orbits of a square matrix was solved. One can easily see that these two problems are closely related. In fact, all matrices in the same similarity orbit have the

---

[*] Received by the editors January 24, 2006; accepted for publication (in revised form) by H. J. Woerdeman November 29, 2006; published electronically April 3, 2007. This work was partially supported by the MEC, Proyecto de Investigación MTM2004-06389-C02-01, and by UPV/EHU, Proyecto de Investigación 9/UPV00100.310-14578/2002.

http://www.siam.org/journals/simax/29-2/65072.html

[†] Departamento de Ciencias de la Computación e IA, Universidad del País Vasco, Apdo 649, 20080 Donostia-San Sebastián, Spain (itziar@bargana@ehu.es; victoria.fernandez@ehu.es).

[‡] Departamento de Matemática Aplicada y EIO, Universidad del País Vasco, Apdo 644, 48080 Bilbao, Spain (ion.zaballa@ehu.es).

same eigenvalues and partial multiplicities, and every matrix in the closure of that orbit is the limit of matrices in the given orbit. Thus the eigenvalues of such a matrix must be the same as the eigenvalues of the matrices in the orbit (continuity of the eigenvalues w.r.t. the matrix), and it is "close enough" to some matrix in the orbit. So, the difference between the two problems is that in the first one the eigenvalues of the near matrices may change, but the conditions on the partial multiplicities in both problems are the same. The point is that in the problem of the closure, since the eigenvalues do not change, the conditions on the partial multiplicities can be translated into divisibility conditions among the invariant factors (see Lemma 4.1). In [6] the result of [3, 12] was generalized to square matrices over $\mathbb{R}$, and a similar question was addressed for the feedback invariants of matrix pairs, i.e., to say, for the controllability indices and Jordan structure of the noncontrollable part (see Lemma 3.1). This result was generalized in [16] to singular pencils of matrices. The controllability indices are one of the many possible invariants of structure of a matrix pair for similarity. Another system of invariants of structure is the one formed by the Hermite indices. These indices appear in several places in the literature. They were first mentioned, with no specific name, in [9, p. 426] and associated to the degrees of the diagonal elements of the denominator of a matrix fraction description of the system. Such a denominator turns to be a polynomial matrix in Hermite form [9, p. 476]. These indices were used in [8] to study the strict system equivalence. And, above all, in [7] the Hermite indices were shown to play a central role in the study of the topological properties of the orbit space of controllable systems under similarity. Actually, in [7] the possible Hermite indices that can be attained by small additive perturbation of a given complex controllable system were characterized. Also, it was shown that such a characterization is not enough, in the real case, to guarantee the existence of a controllable matrix pair as close to a given system as one may want with those Hermite indices. In this paper we provide a new characterization that works for both the real and complex case. In addition, we also consider the other type of invariants of structure for the similarity of matrix pairs: the partial multiplicities of the noncontrollable part of the system.

**2. Notation and preliminary results.** As already stated, $\mathbb{F}$ will denote either the field of real or complex numbers. $\mathbb{F}[s]$ is the ring of polynomials in the indeterminate $s$ with coefficients in $\mathbb{F}$. We will use $\mathbb{F}^{n\times m}$ and $\mathbb{F}[s]^{n\times m}$ to denote the set of $n\times m$ matrices over $\mathbb{F}$ and $\mathbb{F}[s]$, respectively. $Gl_n(\mathbb{F})$ denotes the linear group of order $n$ over $\mathbb{F}$.

We will use Greek letters to denote polynomials. If $\alpha \in \mathbb{F}[s]$ we will write $d(\alpha)$ for the degree of $\alpha$ and $\Lambda(\alpha)$ for the set of roots in $\mathbb{C}$ of the polynomial $\alpha$. If $\alpha \in \mathbb{R}[s]$ we will denote by $r(\alpha)$ the sum of the multiplicities of the real roots of the polynomial $\alpha$. If $\alpha, \beta \in \mathbb{F}[s]$ we will use $\alpha|\beta$ to mean divides.

As usual, we will say that $(A,B)$ is [illegible] if $\operatorname{rank} \mathcal{C}(A,B) = n$, where

$$\mathcal{C}(A,B) := \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \in \mathbb{F}^{n\times nm}$$

is the [illegible] of $(A,B)$.

We will use the [illegible] as introduced in [19]. Given a matrix pair $(A,B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$, the matrix

$$\mathcal{H}(A,B) := [b_1 \ Ab_1 \ \cdots \ A^{n-1}b_1 \ \cdots \ b_m \ Ab_m \ \cdots \ A^{n-1}b_m],$$

where $b_i \in \mathbb{F}^{n\times 1}$ is the $i$th column of $B$, will be called the [illegible] of $(A,B)$.

If rank $\mathcal{H}(A,B)=r$ and we select from left to right the first $r$ linearly independent columns in $\mathcal{H}(A,B)$ and we write them as

$$b_1,\dots,A^{h_1-1}b_1,\dots,b_m,\dots,A^{h_m-1}b_m,$$

then $h_1,\dots,h_m$ are the [illegible] of the pair (we agree that $h_i=0$ if the column $b_i$ has not been selected).

Two matrix pairs $(A,B),(\overline{A},\overline{B})\in\mathbb{F}^{n\times n}\times\mathbb{F}^{n\times m}$ are [illegible], $(A,B)\overset{s}{\sim}(\overline{A},\overline{B})$, if there exists $P\in Gl_n(\mathbb{F})$ such that $\overline{A}=PAP^{-1}$ and $\overline{B}=PB$.

In the following lemma we give a canonical form for similarity associated to the Hermite indices (see [19, 8, 14]).

LEMMA 2.1. [illegible] $(A,B)\in\mathbb{F}^{n\times n}\times\mathbb{F}^{n\times m}$ [illegible] $h_1,\dots,h_m$ [illegible] $P\in GL_n(\mathbb{F})$ [illegible]

$$(PAP^{-1},PB)=\left(\begin{bmatrix}A_{11}&A_{12}&\cdots&A_{1m}\\0&A_{22}&\cdots&A_{2m}\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&A_{mm}\end{bmatrix},\begin{bmatrix}B_{11}&B_{12}&\cdots&B_{1m}\\0&B_{22}&\cdots&B_{2m}\\\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&B_{mm}\end{bmatrix}\right),$$

[illegible] $i=1,\dots,m$ [illegible] $j=i,\dots,m$

(i)

$$A_{ii}=\begin{bmatrix}0&0&\cdots&0&x_{ii0}\\1&0&\cdots&0&x_{ii1}\\0&1&\cdots&0&x_{ii2}\\\vdots&\vdots&\ddots&\vdots&\vdots\\0&0&\cdots&1&x_{iih_i-1}\end{bmatrix}\in\mathbb{F}^{h_i\times h_i},$$

$$A_{ij}=\begin{bmatrix}0&0&\cdots&0&x_{ji0}\\0&0&\cdots&0&x_{ji1}\\0&0&\cdots&0&x_{ji2}\\\vdots&\vdots&\ddots&\vdots&\vdots\\0&0&\cdots&0&x_{jih_i-1}\end{bmatrix}\in\mathbb{F}^{h_i\times h_j},\quad i<j,$$

(ii)

$$\begin{aligned}&B_{ii}=[1\ 0\ \cdots\ 0]^T\in\mathbb{F}^{h_i\times 1} && \text{[illegible]}\ h_i>0,\\&B_{ij}=0\in\mathbb{F}^{h_i\times 1} && \text{[illegible]}\ h_j>0,\quad i<j,\\&B_{ij}=[x_{ji0}\ x_{ji1}\ \cdots\ x_{jih_i-1}]^T\in\mathbb{F}^{h_i\times 1} && \text{[illegible]}\ h_j=0,\quad i\le j,\end{aligned}$$

[illegible] 0 [illegible] 0 [illegible]

If the pair $(A,B)$ is not controllable, then the Hermite indices are also invariant under similarity. Furthermore, if we call invariant factors of $(A,B)$ those of the polynomial matrix $[sI-A\ B]$, then these polynomials are also invariant under similarity. Notice that the invariant factors of $(A,B)$ are equal to 1 if and only if the pair $(A,B)$ is controllable [17].

Throughout this paper, for a given polynomial $\alpha = s^h - a_{h-1}s^{h-1} - \cdots - a_1 s - a_0$, we will say that

$$\begin{bmatrix} 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & \cdots & 0 & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{h-1} \end{bmatrix}$$

is [illegible] of $\alpha$. Thus, in Lemma 2.1, block $A_{ii}$ is the companion matrix of $\theta_i = s^{h_i} - x_{iih_i-1}s^{h_i-1} \cdots - x_{ii1}s - x_{ii0}$, $i = 1, \ldots, m$. These polynomials will be called the [illegible] of $(A, B)$. Actually, as pointed out in [9, p. 476], they are the polynomials appearing on the diagonal of the Hermite normal form of the denominator of any right coprime factorization of the matrix transfer function $(sI_n - A)^{-1}B$.

For noncontrollable pairs we will use the well-known [illegible] (see, for example [9, p. 361]).

Lemma 2.2 (Zaballa [20]). [illegible] $(A, B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$. [illegible]

$$\left( \begin{bmatrix} A_1 & A_2 \\ 0 & A_3 \end{bmatrix}, \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \right) \quad [illegible] \quad \left( \begin{bmatrix} \overline{A}_1 & \overline{A}_2 \\ 0 & \overline{A}_3 \end{bmatrix}, \begin{bmatrix} \overline{B}_1 \\ 0 \end{bmatrix} \right)$$

[illegible] $(A, B)$ [illegible]

$$(A_1, B_1) \overset{s}{\sim} (\overline{A}_1, \overline{B}_1) \quad [illegible] \quad A_3 \overset{s}{\sim} \overline{A}_3.$$

Then, given any Kalman decomposition of a pair $(A, B)$, the controllable part $(A_1, B_1)$ and the square block $A_3$ are determined up to similarity by $(A, B)$. Therefore, the diagonal Hermite polynomials of $(A_1, B_1)$ are invariant under similarity and, in this paper, will be called diagonal Hermite polynomials of $(A, B)$. On the other hand, the invariant factors different from 1 of $(A, B)$ and those of $A_3$ coincide (see [18]). This means, among other things, that if $r = \operatorname{rank} \mathcal{C}(A, B)$ and $\alpha_n \mid \cdots \mid \alpha_1$ are the invariant factors of $(A, B)$, then $\alpha_{n-r+1} = \cdots = \alpha_n = 1$.

Let $(A, B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$; a complex number $\lambda$ is said to be an eigenvalue of $(A, B)$ if $\lambda$ is an eigenvalue of $A_3$ in a Kalman decomposition of $(A, B)$. We will denote by $\Lambda(A, B)$ the set of the eigenvalues of $(A, B)$. Similarly we define the characteristic polynomial of $(A, B)$, the Segre characteristic, and the algebraic multiplicity of $\lambda \in \Lambda(A, B)$. This algebraic multiplicity will be denoted by $m_{(A,B)}(\lambda)$. That is to say, if $\alpha_n | \cdots | \alpha_1$ are the invariant factors of $(A, B)$ and

$$\alpha_j = \prod_{i=1}^{p} (s - \lambda_i)^{s_{ij}}, \quad 1 \le j \le n,$$

then $(s_{i1}, \ldots, s_{in})$ is the Segre characteristic corresponding to $\lambda_i$ and $m_{(A,B)}(\lambda_i) = \sum_{j=1}^{n} s_{ij}$. We agree that if $\lambda \notin \Lambda(A, B)$, then $m_{(A,B)}(\lambda) = 0$ and $(0, 0, \ldots)$ is the partition corresponding to the Segre characteristic of $\lambda$.

Let $(A, [B_1\ B_2]) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times(m_1+m_2)}$. Let $\theta_1, \ldots, \theta_{m_1}, \theta_{m_1+1}, \ldots, \theta_{m_1+m_2}$ be the diagonal Hermite polynomials of $(A, [B_1\ B_2])$, and let $\vartheta$ be the characteristic polynomial of $(A, [B_1\ B_2])$. It is easy to prove that $\theta_1, \ldots, \theta_{m_1}$ are the diagonal Hermite polynomials and $\theta = \vartheta \prod_{i=1}^{m_2} \theta_{m_1+i}$ is the characteristic polynomial of $(A, B_1)$.

Given $A = (a_{ij}) \in \mathbb{F}^{n\times m}$ we are going to consider the following matrix norm:

$$\| A \| = \sum_{i,j} |a_{ij}|.$$

The set $\mathbb{F}^{n\times m}$ is a metric space with the distance associated to this norm. For a given $n$ and a polynomial $\theta = c_n s^n + c_{n-1}s^{n-1} + \cdots + c_1 s + c_0 \in \mathbb{F}[s]$, we will denote

$$\| \theta \| = \sum_{i=0}^{n} |c_i|$$

which is a norm on the vector space of polynomials of degree less than or equal to $n$.

If $z \in \mathbb{C}$ and $r$ is a positive real number, the open ball of $\mathbb{C}$ centered in $\alpha$ and radius $r$ is denoted by $B(z, r)$.

Let $(A, B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$, let $\Lambda(A, B) = \{\lambda_1, \ldots, \lambda_u\}$, and let $\eta$ be a positive real number. We define the $\eta$-neighborhood of the spectrum of $(A, B)$ as the set

$$\mathcal{V}_\eta(A, B) := \dot{\cup}_{i=1}^{u} B(\lambda_i, \eta)$$

whenever the balls $B(\lambda_i, \eta)$, $i = 1, \ldots, u$, are pairwise disjoint.

From now on $\eta$ will always mean a positive real number small enough for the expresion $\eta$-neighborhood of the spectrum of $(A, B)$, or of $A$, to make sense.

Let $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ be two partitions of nonnegative integers with its components arranged in nonincreasing order. Following [13], we will say that $a$ is majorized by $b$, $a \prec b$, if

$$\sum_{j=1}^{k} a_j \le \sum_{j=1}^{k} b_j, \quad k = 1, \ldots, n-1, \text{ and } \sum_{j=1}^{n} a_j = \sum_{j=1}^{n} b_j.$$

Moreover, $a + b$ is the partition whose $i$th component is $a_i + b_i$.

We finish this section with two relevant results of [7] with the terminology of [6].

LEMMA 2.3 (see [7, proof of Thm. 4.2]). $(A, B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$ $h_1, \ldots, h_m$ $\epsilon > 0$ $\| [A\ B] - [A'\ B'] \| < \epsilon$ $h'_1, \ldots, h'_m$ $(A', B')$

$$\sum_{j=1}^{i} h_j \le \sum_{j=1}^{i} h'_j, \qquad i = 1, \ldots, m. \tag{2.1}$$

$(A, B)$ $(A', B')$

$$\sum_{j=1}^{m} h_j = \sum_{j=1}^{m} h'_j. \tag{2.2}$$

LEMMA 2.4 (see [7, Thm. 4.2]). $(A, B) \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times m}$ $h_1, \ldots, h_m$ $h'_1, \ldots, h'_m$ $\epsilon > 0$ $(A', B') \in \mathbb{C}^{n\times n} \times \mathbb{C}^{n\times m}$

(i) $\| [A\ B] - [A'\ B'] \| < \epsilon$,

(ii) $h'_1, \ldots, h'_m$ $(A', B')$,

(2.1) (2.2)

In the present work we generalize the above results for $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$ and for noncontrollable systems. In section 3 we establish conditions that must necessarily satisfy the Hermite indices and the Jordan part of all systems that are close enough to a given system. Then Lemma 2.3 will follow as a particular case. In section 4 we generalize Lemma 2.4 by giving necessary and sufficient conditions for the existence of matrix pairs as close to a given pair as desired and having prescribed its Hermite indices and invariant factors.

**3. Necessary conditions.** First of all we give Theorem 4.3 of [6] in a different form.

LEMMA 3.1. [illegible] $(A,B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$[illegible] $\eta > 0$ [illegible] $\lambda \in \Lambda(A,B)$[illegible] $s$ [illegible] $\lambda$[illegible] $(A,B)$ [illegible] $\epsilon > 0$ [illegible] $\| [A \ B] - [A' \ B'] \| < \epsilon$ [illegible]

(i) $\Lambda(A',B') \subset \mathcal{V}_\eta(A,B)$[illegible]

(ii) [illegible] $\mu_1,\dots,\mu_t$[illegible] $(A',B')$[illegible] $\mathcal{B}(\lambda,\eta)$[illegible] $s'_i$[illegible] $\mu_i$[illegible] $(A',B')$, $i = 1,\dots,t$ [illegible] $q \geq 0$ [illegible]

$$s \prec \sum_{j=1}^{t} s'_i + (q,0,\dots).$$

[illegible] 3.2. Condition (ii) implies

$$\sum_{\mu\in\mathcal{B}(\lambda,\eta)\cap\Lambda(A',B')} m_{(A',B')}(\mu) \leq m_{(A,B)}(\lambda).$$

Moreover, if $(A,B) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$ and $\lambda \in \mathbb{C} \setminus \mathbb{R}$, then $\overline{\mu}_1,\dots,\overline{\mu}_t$ are the eigenvalues of $(A',B')$ in $\mathcal{B}(\overline{\lambda},\eta)$, and $s'_i$ is the partition corresponding to $\overline{\mu}_i$ in the Segre characteristic of $(A',B')$, $i = 1,\dots,t$.

THEOREM 3.3. [illegible] $(A,B) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}$ $\eta > 0$[illegible] $\theta_1,\dots,\theta_m$ [illegible] $(A,B)$ [illegible] $\Lambda(A,B) = \{\lambda_1,\dots,\lambda_p\}$[illegible] $s_i$ [illegible] $\lambda_i$, $i = 1,\dots,p$ [illegible] $(A,B)$ [illegible] $\epsilon > 0$ [illegible] $\| [A \ B] - [A' \ B'] \| < \epsilon$ [illegible]

(i) $\Lambda(A',B') \subset \mathcal{V}_\eta(A,B)$

(ii) [illegible] $\mu_{i1},\dots,\mu_{it_i}$[illegible] $(A',B')$[illegible] $\mathcal{B}(\lambda_i,\eta)$, $i = 1,\dots,p$[illegible] $s'_{ij}$ [illegible] $\mu_{ij}$, $i = 1,\dots,p$, $j = 1,\dots,t_i$ [illegible] $(A',B')$ [illegible] $q_i \geq 0$, $i = 1,\dots,p$ [illegible]

$$s_i \prec \sum_{j=1}^{t_i} s'_{ij} + (q_i,0,\dots), \quad i = 1,\dots,p,$$

(iii) [illegible] $h'_1,\dots,h'_m$[illegible] $(A',B')$ [illegible] $\beta_1,\dots,\beta_m \in \mathbb{F}[s]$ [illegible] $d(\beta_i) = h'_i$, $i = 1,\dots,m$[illegible]

$$\beta_j \cdots \beta_m \mid \theta_j \cdots \theta_m \omega, \quad j = 2,\dots,m,$$

$$\beta_1 \cdots \beta_m = \theta_1 \cdots \theta_m \omega,$$

[illegible] $\omega := \prod_{i=1}^{p} (s-\lambda_i)^{q_i}$.

*Proof.* Let $B = [b_1 \;\cdots\; b_m]$. For $j = 1, \ldots, m$, put $B_j := [b_1 \;\cdots\; b_j]$ and $(A, B_0) := A$.

Let $\theta_{m+1}$ be the characteristic polynomial of $(A, B)$. Then $\Lambda(A, B_j) = \Lambda(\theta_{j+1} \ldots \theta_m \theta_{m+1})$ for $j = 0, \ldots, m$.

Suppose that $\Lambda(A) = \{\lambda_1, \ldots, \lambda_p, \lambda_{p+1}, \ldots, \lambda_{p+q}\},\ q \geq 0$.

For $j = 0, \ldots, m$, $i = 1, \ldots, p+q$, let $m_i^j := m_{(A,B_j)}(\lambda_i)$. Notice that $m_i^j = 0$ if $\lambda_i \notin \Lambda(A, B_j)$. Therefore

$$\theta_{j+1} \cdots \theta_m \theta_{m+1} = \prod_{i=1}^{p+q} (s - \lambda_i)^{m_i^j}, \qquad j = 0, \ldots, m. \tag{3.1}$$

By Lemma 3.1, for $j = 0, \ldots, m$ there exists $\varepsilon_j > 0$ such that if $(A', B_j')$ satisfies $\| [A, B_j] - [A', B_j'] \| < \varepsilon_j$, then

$$\Lambda(A', B_j') \subset \mathcal{V}_\eta(A, B_j), \tag{3.2}$$

$$\sum_{\mu \in \mathcal{B}(\lambda_i, \eta) \cap \Lambda(A', B_j')} m_{(A', B_j')}(\mu) \leq m_i^j, \quad i = 1, \ldots, p+q. \tag{3.3}$$

Moreover, if $\mu_{i1}, \ldots, \mu_{it_i}$ are the eigenvalues of $(A', B_m')$ in $\mathcal{B}(\lambda_i, \eta),\ i = 1, \ldots, p$, and $s_{ij}'$ is the partition corresponding to $\mu_{ij},\ i = 1, \ldots, p,\ j = 1, \ldots, t_i$, in the Segre characteristic of $(A', B_m')$, then there exist nonnegative integers $q_i \geq 0,\ i = 1, \ldots, p$, such that

$$s_i \prec \sum_{j=1}^{t_i} s_{ij}' + (q_i, 0, \ldots), \quad i = 1, \ldots, p.$$

Let $\varepsilon = \min_{0 \leq j \leq m}(\varepsilon_j)$, and let $(A', B')$ be such that $\| [A' \; B'] - [A \; B] \| < \varepsilon$. Then $(A', B')$ satisfies conditions (i) and (ii) of theorem.

Let $B' = [b_1' \;\cdots\; b_m']$, and for $j = 1, \ldots, m$, put $B_j' := [b_1' \;\cdots\; b_j']$ and $(A', B_0') := A'$.

Then, for $j = 0, \ldots, m$, $\| [A \; B_j] - [A' \; B_j'] \| < \varepsilon \leq \varepsilon_j$ and therefore (3.2) and (3.3) hold.

Let $h_1', \ldots, h_m'$ be the Hermite indices, let $\theta_1', \ldots, \theta_m'$ be the diagonal Hermite polynomials and let $\theta_{m+1}'$ be the characteristic polynomial of $(A', B')$. Then $h_j' = d(\theta_j')$, $j = 1, \ldots, m$, and

$$\Lambda(A', B_j') = \Lambda(\theta_{j+1}' \cdots \theta_m' \theta_{m+1}'), \quad j = 0, \ldots, m. \tag{3.4}$$

Let

$$n_i^j := \sum_{\mu \in \mathcal{B}(\lambda_i, \eta) \cap \Lambda(A', B_j')} m_{(A', B_j')}(\mu), \quad i = 1, \ldots, p+q;\ j = 0, \ldots, m.$$

For $j = 1, \ldots, m+1$, we define

$$\delta_j := \prod_{i=1}^{p+q} (s - \lambda_i)^{n_i^{j-1}}.$$

Notice that $\delta_j \in \mathbb{F}[s]$. In fact, if $\mathbb{F} = \mathbb{R}$ and $\lambda_i \in \Lambda(A, B_{j-1}) \setminus \mathbb{R}$, then $\lambda_k = \overline{\lambda}_i \in \Lambda(A, B_{j-1})$, and by Remark 3.2, $n_i^{j-1} = n_k^{j-1}$. Moreover,

$$d(\delta_j) = d(\theta_j' \cdots \theta_{m+1}'),$$

and from (3.1) and (3.3) we have that

$$\delta_j \mid \theta_j \cdots \theta_m \theta_{m+1}, \qquad j = 1, \ldots, m+1.$$

Recall that $n_i^m = 0$ for $i = p+1, \ldots, p+q$ and

$$q_i = m_{(A,B)}(\lambda_i) - \sum_{\mu \in \mathcal{B}(\lambda_i,\eta) \cap \Lambda(A',B')} m_{(A',B')}(\mu) = m_i^m - n_i^m, \qquad i = 1, \ldots, p.$$

This implies that $\omega = \prod_{i=1}^{p} (s - \lambda_i)^{q_i} = \frac{\theta_{m+1}}{\delta_{m+1}}$.

From (3.4) we have that $n_i^j \leq n_i^{j-1}$, $i = 1, \ldots, p+q$; $j = 1, \ldots, m$. Therefore $\delta_{j+1} \mid \delta_j$, $j = 1, \ldots, m$. Let

$$\beta_j := \frac{\delta_j}{\delta_{j+1}}, \quad j = 1, \ldots, m.$$

Then $\beta_j \in \mathbb{F}[s]$, $j = 1, \ldots, m$, and

$$\beta_j \cdots \beta_m = \frac{\delta_j}{\delta_{m+1}} \mid \theta_j \cdots \theta_m \frac{\theta_{m+1}}{\delta_{m+1}} = \theta_j \cdots \theta_m \omega, \qquad j = 1, \ldots, m.$$

We have to prove only that $d(\beta_j) = h'_j$, $j = 1, \ldots, m$, and $\beta_1 \cdots \beta_m = \theta_1 \cdots \theta_m \omega$. In fact,

$$\sum_{i=j}^{m} h'_i = d(\theta'_j \cdots \theta'_m) = d(\delta_j) - d(\theta'_{m+1}) = d(\delta_j) - d(\delta_{m+1}) = \sum_{i=j}^{m} d(\beta_i), \quad j = 1, \ldots, m.$$

As a consequence $d(\beta_j) = h'_j$, $j = 1, \ldots, m$.

Finally,

$$d(\beta_1 \cdots \beta_m) = \sum_{i=1}^{m} h'_i = n - d(\theta'_{m+1}) = \sum_{i=1}^{m} h_i + d(\theta_{m+1}) - d(\delta_{m+1}) = d(\theta_1 \theta_2 \cdots \theta_m) + d(\omega).$$

Since $\beta_1 \cdots \beta_m \mid \theta_1 \cdots \theta_m \omega$ we conclude that

$$\beta_1 \cdots \beta_m = \theta_1 \cdots \theta_m \omega$$

as desired. ☐

For the controllable case we have the following corollary.

COROLLARY 3.4. [illegible] $(A, B) \in \mathbb{F}^{n \times n} \times \mathbb{F}^{n \times m}$ [illegible] $\theta_1, \ldots, \theta_m$ [illegible] $\epsilon > 0$ [illegible] $\| [A\ B] - [A'\ B'] \| < \epsilon$ [illegible] $h'_1, \ldots, h'_m$ [illegible] $(A', B')$ [illegible] $\beta_1, \ldots, \beta_m$ [illegible] $d(\beta_i) = h'_i$, $i = 1, \ldots, m$ [illegible]

$$\beta_j \cdots \beta_m \mid \theta_j \cdots \theta_m, \quad j = 2, \ldots, m,$$

$$\beta_1 \cdots \beta_m = \theta_1 \cdots \theta_m.$$

[illegible] 3.5. It is easily seen that, if $\mathbb{F} = \mathbb{C}$, the condition of the previous corollary is equivalent to those of Lemma 2.3.

**4. Sufficient conditions.** This section is devoted to proving the near-converse of Theorem 3.3. The proof is constructive and it gives us a general method to obtain a pair as close to a given pair as desired and with prescribed structure.

Before going into formal proofs let us describe the perturbation process for the controllable case. Given a pair $(A, B)$ with diagonal Hermite polynomials $\theta_1, \ldots, \theta_m$ and given $m$ nonnegative integers $h'_1, \ldots, h'_m$, we have to prove that the condition of the Corollary 3.4 is sufficient for the existence of a controllable pair $(A', B')$ as close as we want to $(A, B)$ and with Hermite indices $h'_1, \ldots, h'_m$. Observe that this condition implies (2.1) and (2.2).

Without loss of generality, we can assume that $(A, B)$ is in the canonical form shown in Lemma 2.1:

$$(A,B) = \left( \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ 0 & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{mm} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ 0 & B_{22} & \cdots & B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B_{mm} \end{bmatrix} \right).$$

The proof is inductive, and the first step is to decrease the index $h_m$ by increasing $h_{m-1}$ in order to obtain a pair $(A_1, B_1)$ with $h_1, \ldots, h_{m-2}, h_{m-1} + h_m - h'_m, h'_m$ as Hermite indices. This will be done by putting the coefficients of the polynomial $\beta_m$ multiplied by an appropriate scalar in the last column of the $(m, m-1)$ block:

$$(A_1,B_1) = \left( \begin{bmatrix} A_{11} & \cdots & A_{1,m-1} & A_{1m} \\ 0 & \cdots & A_{2,m-1} & A_{2m} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & A_{m-1,m-1} & A_{m-1,m} \\ 0 & \cdots & C & A_{mm} \end{bmatrix}, \begin{bmatrix} B_{11} & \cdots & B_{1m} \\ 0 & \cdots & B_{2m} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & B_{m-1,m} \\ 0 & \ldots & B_{mm} \end{bmatrix} \right).$$

This pair is similar to

$$\left( \begin{bmatrix} A_{11} & \cdots & \overline{A}_{1,m-1} & \overline{A}_{1m} \\ 0 & \cdots & \overline{A}_{2,m-1} & \overline{A}_{2m} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \overline{A}_{m-1,m-1} & \overline{A}_{m-1,m} \\ 0 & \cdots & 0 & \overline{A}_{mm} \end{bmatrix}, \begin{bmatrix} B_{11} & \cdots & B_{1,m-1} & B_{1m} \\ 0 & \cdots & B_{2,m-1} & B_{2m} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \overline{B}_{m-1,m-1} & \overline{B}_{m-1,m} \\ 0 & \ldots & 0 & \overline{B}_{mm} \end{bmatrix} \right),$$

where $\overline{A}_{m-1,m-1} \in \mathbb{F}^{(h_{m-1}+h_m-h'_m)\times(h_{m-1}+h_m-h'_m)}$ and $\overline{A}_{m,m} \in \mathbb{F}^{h_m \times h_m}$.

Now, if we are perturbing in $\mathbb{C}$, as $h'_{m-1} \leq h_{m-1} + h_m - h'_m$, we can continue with the induction process and obtain a pair $(A_2, B_2)$ with Hermite indices $h_1, \ldots, h_{m-3}, h_{m-2} + h_{m-1} + h_m - h'_m - h'_{m-1}, h'_{m-1}, h'_m$ and so on.

But, if we are perturbing in $\mathbb{R}$ and $\theta_1, \ldots, \theta_{m-2}, \bar{\theta}_{m-1}, \bar{\theta}_m$ are the diagonal Hermite polynomials of $(A_1, B_1)$, in order to continue with the induction process we need to guarantee the existence of a polynomial $\bar{\beta}_{m-1} \in \mathbb{R}[s]$ such that $d(\bar{\beta}_{m-1}) = h'_{m-1}$ and $\bar{\beta}_{m-1} \mid \bar{\theta}_{m-1}$. As divisibility conditions are involved, we will have to take into account the sum of the multiplicities of the roots of $\bar{\theta}_{m-1}$ in $\mathbb{R}$, $r(\bar{\theta}_{m-1})$.

On the one hand, notice that $\bar{\theta}_m = \beta_m$, and as $\beta_{m-1}\beta_m \mid \theta_{m-1}\theta_m$, we have that there exists a polynomial $\nu$ such that $\frac{\theta_{m-1}\theta_m}{\beta_m} = \beta_{m-1}\nu$.

On the other hand, $\bar{\theta}_{m-1} = \frac{\theta_{m-1}\theta_m - \beta_m\psi}{\beta_m}$, where $\psi$ is a polynomial such that $d(\psi) < d(\theta_{m-1})$. Then, $\bar{\theta}_{m-1} = \beta_{m-1}\nu - \psi$.

Now, if $r(\bar{\theta}_{m-1}) = r(\beta_{m-1}\nu)$, we can guarantee the existence of $\bar{\beta}_{m-1}$ such that $\bar{\beta}_{m-1} \mid \bar{\theta}_{m-1}$ and $d(\bar{\beta}_{m-1}) = d(\beta_{m-1})$. If $\psi \neq 0$, i.e., if the block $A_{m-1,m} \neq 0$, the mentioned equality cannot be assured.

Therefore, we first perturb in the block $A_{m-1,m-1}$ for the spectrum of this block and that of $A_{m,m}$ to be disjoint and to put a zero in the block $(m-1,m)$ by similarity.

In the noncontrollable case our first goal is to modify the Hermite indices but prescribing the spectrum of $(A',B')$. This will be done in Theorem 4.11. In this case we start by modifying the last Hermite index to obtain a pair $(\bar{A},\bar{B})$ with the prescribed invariant factors, Hermite indices $h_1,\dots,h_{m-1},h_m+d(\omega)$, and diagonal Hermite polynomials $\theta_1,\dots,\theta_{m-1},\bar{\theta}_m$. As in the controllable case, when $\mathbb{F}=\mathbb{R}$ we can continue with the process if $r(\bar{\theta}_m) = r(\theta_m\omega)$.

In the following example we will explain the steps for the controllable case.

1. Let

$$(A,B) = \left( \left[ \begin{array}{cc|c|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right] \right).$$

This pair has $h_1 = 2, h_2 = 1, h_3 = 2$ as Hermite indices and $\theta_1 = s^2, \theta_2 = s$, $\theta_3 = s^2$ as diagonal Hermite polynomials.

Our goal is to obtain a pair $(A',B')$ as close to $(A,B)$ as desired and with Hermite indices $h_1' = 3, h_2' = 1, h_3' = 1$. If we chose $\beta_3 = s$, $\beta_2 = s$, and $\beta_1 = s^3$, we can verify that the conditions of Corollary 3.4 are satisfied:

$$\beta_3 \mid \theta_3, \ \beta_2\beta_3 \mid \theta_2\theta_3 \text{ and } \beta_1\beta_2\beta_3 = \theta_1\theta_2\theta_3.$$

As we have said, we first modify the last Hermite index to obtain a pair with $h_3' = 1$ putting in the last column of the block $A_{32}$ the coefficients of $\beta_3$ multiplied by a convenient $\epsilon_1$:

$$\left( \left[ \begin{array}{cc|c|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \epsilon_1 & 1 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right] \right).$$

This pair has Hermite indices $2,2,1$ and diagonal Hermite polynomials $\overline{\theta}_1 = s^2, \overline{\theta}_2 = s^2 + \epsilon_1$, $\overline{\theta}_3 = s$, and it is similar to

$$(\overline{A},\overline{B}) = \left( \left[ \begin{array}{cc|cc|c} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -\epsilon_1 & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{\epsilon_1} \\ \hline 0 & 0 & 0 & 0 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array} \right] \right).$$

To continue with the process we consider the polynomial $\overline{\beta}_2 = s - \sqrt{\epsilon_1} i$ of degree $h_2'$ that divides $\overline{\theta}_2$, and we put in the last column of the block $A_{22}$ its coefficients multiplied by a convenient $\epsilon_2 > 0$:

$$(\overline{A}', \overline{B}') = \left( \left[ \begin{array}{cc|cc|c} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & -\sqrt{\epsilon_1} i \epsilon_2 & 0 & -\epsilon_1 & 0 \\ 0 & \epsilon_2 & 1 & 0 & \frac{1}{\epsilon_1} \\ \hline 0 & 0 & 0 & 0 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array} \right] \right).$$

This pair has Hermite indices $h'_1 = 3, h'_2 = 1, h'_3 = 1$, but notice that it is not a real pair. In this case $\overline{\theta}_2$ has no real roots, and therefore, there is no $\overline{\beta}_2 \in \mathbb{R}[s]$ of degree 1 such that $\overline{\beta}_2 | \overline{\theta}_2$. That is because $r(\overline{\theta}_2) \neq r(\frac{\theta_2 \theta_3}{\beta_3})$.

Then, in the real case we first perturb matrix $A$ for the spectrum of blocks $A_{22}$ and $A_{33}$ to be disjoint, and by similarity transformations we can put a zero in the block $A_{23}$:

$$\left( \left[ \begin{array}{cc|c|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \epsilon_0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right] \right).$$

This pair is similar to

$$(\overline{A}, \overline{B}) = \left( \left[ \begin{array}{cc|c|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \epsilon_0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & -\frac{1}{\epsilon_0^2} \\ \hline 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right] \right).$$

Now we proceed as before. First we obtain $h'_3$ by perturbing in the block $A_{32}$:

$$\left( \left[ \begin{array}{cc|c|cc} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \epsilon_0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \epsilon_1 & 1 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & -\frac{1}{\epsilon_0^2} \\ \hline 0 & 0 & 1 \\ 0 & 0 & 0 \end{array} \right] \right).$$

This pair has $2, 2, 1$ as Hermite indices and $\overline{\overline{\theta}}_1 = s^2, \overline{\overline{\theta}}_2 = s^2 - \epsilon_0 s$, and $\overline{\overline{\theta}}_3 = s$ as diagonal Hermite polynomials, and it is similar to

$$(\overline{\overline{A}}, \overline{\overline{B}}) = \left( \left[ \begin{array}{cc|cc|c} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -\frac{\epsilon_0}{\epsilon_1} \\ 0 & 0 & 1 & \epsilon_0 & -\frac{1}{\epsilon_0^2} + \frac{1}{\epsilon_1} \\ \hline 0 & 0 & 0 & 0 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array} \right] \right).$$

Now $r(\overline{\overline{\theta}}_2) = r(\frac{\theta_2 \theta_3}{\beta_3})$. Then, we can choose $\overline{\overline{\beta}}_2 = s \in \mathbb{R}[s]$ of degree $h'_2$ to put its coefficients multiplied by a convenient $\epsilon_2 > 0$ in the last column of the block $\overline{\overline{A}}_{21}$:

$$(\overline{\overline{A}}', \overline{\overline{B}}') = \left( \left[ \begin{array}{cc|cc|c} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -\frac{\epsilon_0}{\epsilon_1} \\ 0 & \epsilon_2 & 1 & \epsilon_0 & -\frac{1}{\epsilon_0^2} + \frac{1}{\epsilon_1} \\ \hline 0 & 0 & 0 & 0 & 0 \end{array} \right] \left[ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 1 \end{array} \right] \right).$$

To finish we must undo the two similarity transformations that we have done throughout the process to obtain

$$(A',B')=\left(\left[\begin{array}{cc|c|cc} 0 & 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0\\ \hline 0 & \frac{\epsilon_2(\epsilon_0^2+\epsilon_1)}{\epsilon_0} & \frac{\epsilon_0^2+\epsilon_1}{\epsilon_0} & -\frac{\epsilon_1}{\epsilon_0^3} & -1-\frac{\epsilon_1}{\epsilon_0^2}\\ \hline 0 & 0 & 0 & 0 & 0\\ 0 & \epsilon_1\epsilon_2 & \epsilon_1 & 1-\frac{\epsilon_1}{\epsilon_0^2} & -\frac{\epsilon_1}{\epsilon_0} \end{array}\right]\left[\begin{array}{ccc} 1 & 0 & 0\\ 0 & 0 & 0\\ \hline 0 & 1 & 0\\ \hline 0 & 0 & 1\\ 0 & 0 & 0 \end{array}\right]\right).$$

This pair is a real pair and has the prescribed Hermite indices.

Now, if we choose $\epsilon_2=\epsilon_0$ and $\epsilon_1=\epsilon_0^4$ it is easy to verify that

$$\parallel [A\ B]-[A'\ B'] \parallel = \epsilon_0^5+2\epsilon_0^4+2\epsilon_0^3+3\epsilon_0^2+2\epsilon_0.$$

By choosing $\epsilon_0$ small enough, $(A',B')$ can be as close as we want to $(A,B)$.

Lemma 4.1 (see [2]). [illegible] $A\in\mathbb{F}^{n\times n}$ [illegible] $\alpha_n\mid\dots\mid\alpha_1$[illegible] $\alpha'_1,\dots,\alpha'_n\in\mathbb{F}[s]$ [illegible] $\alpha'_n\mid\dots\mid\alpha'_1$ [illegible] $\epsilon>0$ [illegible] $A'\in\mathbb{F}^{n\times n}$ [illegible]

(i) $\parallel A'-A\parallel<\epsilon$

(ii) $\alpha'_1,\dots,\alpha'_n$[illegible] $A'$ [illegible]

$$\alpha_1\cdots\alpha_k\mid\alpha'_1\cdots\alpha'_k,\qquad k=1,\dots,n-1,$$

$$\alpha_1\cdots\alpha_n=\alpha'_1\cdots\alpha'_n.$$

The aim of the next lemma, whose proof is straightforward, is to show that the pair $(A,B)$ can be replaced by any pair in its similarity class.

Lemma 4.2. [illegible] $(A,B),(\overline{A},\overline{B})\in\mathbb{F}^{n\times n}\times\mathbb{F}^{n\times m}$ [illegible] $(A,B)\overset{s}{\sim}(\overline{A},\overline{B})$[illegible] $\mathcal{C}_s$ [illegible] $\mathbb{F}^{n\times n}\times\mathbb{F}^{n\times m}$. [illegible]

(1) [illegible] $\epsilon>0$ [illegible] $(A',B')\in\mathcal{C}_s$ [illegible]

$$\parallel [A\ B]-[A'\ B'] \parallel<\epsilon.$$

(2) [illegible] $\epsilon'>0$ [illegible] $(\overline{A}',\overline{B}')\in\mathcal{C}_s$ [illegible]

$$\parallel [\overline{A}\ \overline{B}]-[\overline{A}'\ \overline{B}'] \parallel<\epsilon'.$$

[illegible] 4.3. A similar lemma can be made if we prescribe some of the invariants for similarity instead of the whole similarity class.

Lemma 4.4. [illegible]

$$(A,b)=\left(\begin{bmatrix} A_1 & 0\\ 0 & J\end{bmatrix},\begin{bmatrix} b_1\\ 0\end{bmatrix}\right)\in\mathbb{F}^{n\times n}\times\mathbb{F}^{n\times 1},$$

[illegible] $(A_1,b_1)\in\mathbb{F}^{h\times h}\times\mathbb{F}^{h\times 1}$[illegible] $\theta$[illegible] $d(\theta)=h$[illegible] $J$[illegible] $\alpha$

[illegible] $\alpha',\omega\in\mathbb{F}[s]$ [illegible] $\alpha=\alpha'\omega$

[illegible] $\epsilon>0$ [illegible] $(A',b')\in\mathbb{F}^{n\times n}\times\mathbb{F}^{n\times 1}$ [illegible]

(i) $\parallel [A'\ b']-[A\ b] \parallel<\epsilon$

(ii) [illegible] $(A',b')$[illegible] $\theta'=\theta\omega$

(iii) [illegible] $(A', b')$ [illegible] $\alpha'$ [illegible] $\alpha' = 1$ [illegible] $(A', b')$ [illegible]

[illegible]. Let $\epsilon > 0$. If $\alpha' = \alpha$, we put $(A', b') := (A, b)$.

If $\alpha' = s^d - a_{d-1}s^{d-1} - \cdots - a_1 s - a_0$ with $d < n - h = d(\alpha)$, we define

$$c := \epsilon_1 \left[ \begin{array}{cccccccc} a_0 & a_1 & \cdots & a_{d-1} & -1 & 0 & \cdots & 0 \end{array} \right]^T \in \mathbb{F}^{(n-h)\times 1},$$

$0 < \epsilon_1 < \epsilon / \sum_{i=0}^{d-1} |a_i| + 1$.

[illegible] 1. If $h = 0$, (i.e., $(A_1, b_1)$ is absent), then we put

$$(A', b') := (J, c) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times 1}.$$

[illegible] 2. If $h > 0$, then we put $C := \left[ \begin{array}{cc} 0 & c \end{array} \right] \in \mathbb{F}^{(n-h)\times h}$ and

$$(A', b') := \left( \left[ \begin{array}{cc} A_1 & 0 \\ C & J \end{array} \right], \left[ \begin{array}{c} b_1 \\ 0 \end{array} \right] \right) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times 1}.$$

It is easily seen that the only nontrivial invariant factor of $(A', b')$ is $\alpha'$.

The characteristic polynomial of $A'$ is $\theta\alpha$. If the diagonal Hermite polynomial of $(A', b')$ is $\theta'$, then $\theta\alpha = \theta'\alpha'$, and so $\theta' = \theta\omega$. □

The following lemma can be easily proved (see [4, p. 225] or [10, p. 422]).

LEMMA 4.5. [illegible]

$$(A, B) = \left( \left[ \begin{array}{cc} A_1 & X \\ 0 & J \end{array} \right], \left[ \begin{array}{c} B_1 \\ 0 \end{array} \right] \right) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times m}.$$

[illegible] $\Lambda(A_1) \cap \Lambda(J) = \emptyset$ [illegible]

$$(A, B) \overset{s}{\sim} \left( \left[ \begin{array}{cc} A_1 & 0 \\ 0 & J \end{array} \right], \left[ \begin{array}{c} B_1 \\ 0 \end{array} \right] \right).$$

LEMMA 4.6. [illegible] $\theta \in \mathbb{R}[s]$ [illegible] $S = \{\lambda_1, \ldots, \lambda_p\} \subset \mathbb{C}$ [illegible] $\epsilon > 0$ [illegible] $\vartheta \in \mathbb{R}[s]$ [illegible]

$$\| \vartheta - \theta \| < \epsilon, \quad d(\vartheta) = d(\theta), \quad \Lambda(\vartheta) \cap S = \emptyset, \quad r(\vartheta) = r(\theta),$$

[illegible] $r(\theta)$ [illegible] $\theta$

[illegible]. If $\theta = \prod_{i=1}^{n}(s - \mu_i)$, let $\vartheta := \prod_{i=1}^{n}(s - \mu_i - r)$, $r \in \mathbb{R}$. For any $r \in \mathbb{R}$ we have that $\vartheta$ is a monic polynomial, $\vartheta \in \mathbb{R}[s]$, $d(\vartheta) = n$, and $r(\vartheta) = r(\theta)$. By the continuity of the coefficients of a polynomial w.r.t. its roots it is easily seen that it is possible to choose $r$ such that conditions $\| \vartheta - \theta \| < \epsilon$ and $\Lambda(\vartheta) \cap S = \emptyset$ hold. □

LEMMA 4.7. [illegible] $(A, b) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times 1}$ [illegible] $\theta$ [illegible] $\alpha_n \mid \cdots \mid \alpha_1$ [illegible] $\alpha'_1, \ldots, \alpha'_n, \omega \in \mathbb{R}[s]$ [illegible] $\alpha'_n \mid \cdots \mid \alpha'_1$ [illegible]

$$\alpha_1 \cdots \alpha_k \mid \alpha'_1 \cdots \alpha'_k \omega, \qquad k = 1, \ldots, n-1,$$

$$\alpha_1 \cdots \alpha_n = \alpha'_1 \cdots \alpha'_n \omega,$$

[illegible] $\epsilon > 0$ [illegible] $(A', b') \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times 1}$ [illegible] $\vartheta$ [illegible] $r(\vartheta) = r(\theta)$ [illegible] $d(\vartheta) = d(\theta)$ [illegible]

(i) $\| [A' \; b'] - [A \; b] \| < \epsilon$,

(ii) [illegible] $(A', b')$ [illegible] $\vartheta\omega$

(iii) [illegible] $(A', b')$ [illegible] $\alpha'_1, \ldots, \alpha'_n$.

[illegible] Let $h := d(\theta)$. Then $\alpha_i = \alpha'_i = 1$, $i = n-h+1, \dots, n$.

By Lemma 4.2 we can assume that the pair $(A, b)$ is in Kalman form

$$(A,b) = \left( \begin{bmatrix} A_{11} & X \\ 0 & J \end{bmatrix}, \begin{bmatrix} B_{11} \\ 0 \end{bmatrix} \right),$$

where $(A_{11}, B_{11}) \in \mathbb{R}^{h \times h} \times \mathbb{R}^{h \times 1}$ with $\theta$ as diagonal Hermite polynomial, $d(\theta) = h$, $A_{11}$ is the companion matrix of the polynomial $\theta$, and $\alpha_{n-h} \mid \cdots \mid \alpha_1$ are the invariant factors of $J$. Remember that if $h = 0$, then $(A_{11}, B_{11})$ and $X$ vanish.

Let $\alpha''_k := \alpha'_k$, $k = 2, \dots, n-h$, and $\alpha''_1 := \alpha'_1\omega$, then

$$\alpha_1 \cdots \alpha_k \mid \alpha''_1 \cdots \alpha''_k, \qquad k = 1, \dots, n-h,$$

$$\alpha_1 \cdots \alpha_{n-h} = \alpha''_1 \cdots \alpha''_{n-h}.$$

Let $\epsilon > 0$. By Lemma 4.1 there exists a matrix $\overline{J} \in \mathbb{R}^{(n-h)\times(n-h)}$ such that $\| \overline{J} - J \| < \frac{\epsilon}{3}$ and $\alpha''_1, \dots, \alpha''_{n-h}$ are the invariant factors of $\overline{J}$. Then

$$\overline{J} \overset{s}{\sim} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix}, \tag{4.1}$$

where $J_1$ is the companion matrix of $\alpha''_1 = \alpha'_1\omega$.

On the other hand, by Lemma 4.6, there exists a monic polynomial $\vartheta \in \mathbb{R}[s]$ such that $\| \vartheta - \theta \| < \frac{\epsilon}{3}$, $r(\vartheta) = r(\theta)$, $d(\vartheta) = d(\theta)$, and $\Lambda(\vartheta) \cap \Lambda(J) = \emptyset$.

Let $\overline{A}_{11}$ be the companion matrix of $\vartheta$. Then $\| \overline{A}_{11} - A_{11} \| < \frac{\epsilon}{3}$. Put

$$\overline{A} := \begin{bmatrix} \overline{A}_{11} & X \\ 0 & \overline{J} \end{bmatrix}.$$

Then by (4.1) and Lemma 4.5,

$$(\overline{A}, b) \overset{s}{\sim} \left( \begin{bmatrix} \overline{A}_{11} & 0 & 0 \\ 0 & J_1 & 0 \\ 0 & 0 & J_2 \end{bmatrix}, \begin{bmatrix} B_{11} \\ 0 \\ 0 \end{bmatrix} \right).$$

By Lemmas 4.2 and 4.4 there exists a matrix pair $(A', b') \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1}$ such that $\| [A' \ b'] - [\overline{A} \ b] \| < \frac{\epsilon}{3}$, the diagonal Hermite polynomial of $(A', b')$ is $\vartheta\omega$, and $\alpha'_1, \dots, \alpha'_n$ are the invariant factors of $(A', b')$. ☐

COROLLARY 4.8. [illegible] $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$ [illegible] $\theta_1, \dots, \theta_m$ [illegible] $\alpha_n \mid \cdots \mid \alpha_1$ [illegible] $\alpha'_1, \dots, \alpha'_n, \omega \in \mathbb{R}[s]$ [illegible] $\alpha'_n \mid \cdots \mid \alpha'_1$ [illegible]

$$\alpha_1 \cdots \alpha_k \mid \alpha'_1 \cdots \alpha'_k\omega, \qquad k = 1, \dots, n-1,$$

$$\alpha_1 \cdots \alpha_n = \alpha'_1 \cdots \alpha'_n\omega,$$

[illegible] $\epsilon > 0$ [illegible] $(A', B') \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$ [illegible] $\vartheta_m$ [illegible] $r(\vartheta_m) = r(\theta_m)$ [illegible] $d(\vartheta_m) = d(\theta_m)$ [illegible]

(i) $\| [A' \ B'] - [A \ B] \| < \epsilon$,

(ii) [illegible] $(A', B')$ [illegible] $\theta_1, \dots, \theta_{m-1}, \vartheta_m\omega$

(iii) [illegible] $(A', B')$ [illegible] $\alpha'_1, \dots, \alpha'_n$.

In the next theorem we solve the controllable case.

THEOREM 4.9. [illegible] $(A,B) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$ [illegible] $\theta_1,\dots,\theta_m$ [illegible] $(A,B)$ [illegible] $h'_1,\dots,h'_m$ [illegible]

[illegible] $\epsilon > 0$ [illegible] $(A',B') \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$ [illegible]

(i) $\| \, [A \ B] - [A' \ B'] \, \| < \epsilon$

(ii) $(A',B')$ [illegible] $h'_1,\dots,h'_m$ [illegible] $\beta_1,\dots,\beta_m \in \mathbb{R}[s]$ [illegible] $d(\beta_i) = h'_i, \ i = 1,\dots,m$ [illegible]

$$\beta_i \cdots \beta_m \mid \theta_i \cdots \theta_m, \quad i = 2,\dots,m,$$
$$\beta_1 \cdots \beta_m = \theta_1 \cdots \theta_m.$$

[illegible]. The necessity is a consequence of Corollary 3.4.

We will prove the sufficiency by induction on $m$. If $m = 1$ the theorem is trivially true with $(A',B') := (A,B)$.

Suppose that the theorem is true up to $m-1$.

By Lemma 2.3 there exists $\epsilon_0 > 0$ such that if $\| \, [A' \ B'] - [A \ B] \, \| < \epsilon_0$, then $(A',B')$ is controllable.

If $B = [b_1 \ \dots \ b_{m-1} \ b_m]$ and $B_{m-1} = [b_1 \ \dots \ b_{m-1}]$ then the matrix pair $(A, B_{m-1})$ has $\theta_1,\dots,\theta_{m-1}$ as diagonal Hermite polynomials and $\alpha_1 = \theta_m, \alpha_2 = \cdots = \alpha_n = 1$ as invariant factors.

Given $\epsilon > 0$, let $\epsilon_1 = \min\{\frac{\epsilon}{2}, \epsilon_0\}$, $\alpha'_n = \cdots = \alpha'_2 := 1$, $\alpha'_1 := \beta_m$, and $\omega := \frac{\theta_m}{\beta_m}$. Then we have that

$$\alpha_1 \cdots \alpha_k = \alpha'_1 \cdots \alpha'_k \omega, \qquad k = 1,\dots,n$$

By Corollary 4.8 there exists a matrix pair $(\overline{A}, \overline{B}_{m-1}) \in \mathbb{F}^{n\times n} \times \mathbb{F}^{n\times(m-1)}$ and a polynomial $\vartheta_{m-1}$ satisfying $r(\vartheta_{m-1}) = r(\theta_{m-1})$ and $d(\vartheta_{m-1}) = d(\theta_{m-1})$ such that $\| \, [\overline{A} \ \overline{B}_{m-1}] - [A \ B_{m-1}] \, \| < \epsilon_1$, $\alpha'_1,\dots,\alpha'_n$ are the invariant factors of $(\overline{A}, \overline{B}_{m-1})$ and its diagonal Hermite polynomials are $\theta_1,\dots,\theta_{m-2}, \vartheta_{m-1}\omega$. Notice that the characteristic polynomial of $(\overline{A}, \overline{B}_{m-1})$ is $\beta_m$.

Put $\bar{\theta}_i := \theta_i$, $i = 1,\dots,m-2$, $\bar{\theta}_{m-1} =: \vartheta_{m-1}\omega$, and $\bar{\theta}_m =: \beta_m$, and let $(\overline{A}, \overline{B}) =: (\overline{A}, [\overline{B}_{m-1} \ b_m])$. Then $(\overline{A}, \overline{B})$ is controllable, and $\bar{\theta}_1,\dots,\bar{\theta}_m$ are its diagonal Hermite polynomials.

By Lemma 2.1, we can write

$$(\overline{A}, \overline{B}) \sim \left( \begin{bmatrix} \overline{A}_1 & X \\ 0 & A'_{mm} \end{bmatrix}, \begin{bmatrix} \overline{B}_1 & Y \\ 0 & B'_{mm} \end{bmatrix} \right),$$

where $(\overline{A}_1, \overline{B}_1) \in \mathbb{R}^{(n-h'_m)\times(n-h'_m)} \times \mathbb{R}^{(n-h'_m)\times(m-1)}$ is a controllable pair with diagonal Hermite polynomials $\bar{\theta}_1,\dots,\bar{\theta}_{m-2},\bar{\theta}_{m-1}$ and $(A'_{mm}, B'_{mm}) \in \mathbb{R}^{h'_m\times h'_m} \times \mathbb{R}^{h'_m\times 1}$ is a controllable pair with $\beta_m$ as diagonal Hermite polynomial.

Bearing in mind that

$$\beta_i \cdots \beta_{m-1} \mid \theta_i \cdots \theta_{m-1}\omega, \quad i = 2,\dots,m-1,$$
$$\beta_1 \cdots \beta_{m-1} = \theta_1 \cdots \theta_{m-1}\omega$$

and that $d(\bar{\theta}_{m-1}) = d(\theta_{m-1}\omega)$ and $r(\bar{\theta}_{m-1}) = r(\theta_{m-1}\omega)$, it is easily seen that there exist $\beta'_i \in \mathbb{R}[s]$, $i = 1,\dots,m-1$, such that $d(\beta'_i) = h'_i, i = 1,\dots,m-1$, and

$$\beta'_i \cdots \beta'_{m-1} \mid \bar{\theta}_i \cdots \bar{\theta}_{m-1}, \quad i = 2,\dots,m,$$
$$\beta'_1 \cdots \beta'_{m-1} = \bar{\theta}_1 \cdots \bar{\theta}_{m-1}.$$

By the induction hypothesis, in any neighborhood of $(\overline{A}_1, \overline{B}_1)$ there exists a controllable matrix pair $(A'_1, B'_1) \in \mathbb{R}^{(n-h'_m)\times(n-h'_m)} \times \mathbb{R}^{(n-h'_m)\times(m-1)}$ with Hermite indices $h'_1, \dots, h'_{m-1}$. Then

$$\left( \begin{bmatrix} A'_1 & X \\ 0 & A'_{mm} \end{bmatrix}, \begin{bmatrix} B'_1 & Y \\ 0 & B'_{mm} \end{bmatrix} \right)$$

has $h'_1, \dots, h'_m$ as Hermite indices. By Lemma 4.2, there exists a matrix pair $(A', B') \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$ such that $\| \, [A' \; B'] - [\overline{A} \; \overline{B}] \, \| < \frac{\epsilon}{2}$ with Hermite indices $h'_1, \dots, h'_m$. ☐

[illegible] 4.10. In the complex case, the divisibility condition of the previous theorem is equivalent to conditions (2.1) and (2.2). Thus, for $\mathbb{F} = \mathbb{C}$ the above theorem and Lemma 2.4 coincide.

THEOREM 4.11. [illegible] $(A, B) \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$ [illegible] $\theta_1, \dots, \theta_m$, [illegible] $\alpha_n \mid \cdots \mid \alpha_1$ [illegible] $\alpha'_1, \dots, \alpha'_n, \omega \in \mathbb{R}[s]$ [illegible] $\alpha'_n \mid \cdots \mid \alpha'_1$ [illegible] $h'_1, \dots, h'_m$ [illegible]

[illegible] $\epsilon$ [illegible] $(A', B') \in \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$ [illegible]

(i) $\| \, [A' \; B'] - [A \; B] \, \| < \epsilon$

(ii) $h'_1, \dots, h'_m$ [illegible] $(A', B')$

(iii) $\alpha'_1, \dots, \alpha'_n$ [illegible] $(A', B')$

[illegible]

(a)

$$\alpha_1 \cdots \alpha_k \mid \alpha'_1 \cdots \alpha'_k \omega, \qquad k = 1, \dots, n-1,$$
$$\alpha_1 \cdots \alpha_n = \alpha'_1 \cdots \alpha'_n \omega;$$

(b) [illegible] $\beta_1, \dots, \beta_m \in \mathbb{R}[s]$ [illegible] $d(\beta_j) = h'_j$, $j = 1, \dots, m$ [illegible]

$$\beta_k \cdots \beta_m \mid \theta_k \cdots \theta_m \omega, \quad k = 2, \dots, m,$$
$$\beta_1 \cdots \beta_m = \theta_1 \cdots \theta_m \omega.$$

[illegible]. Conditions (a) and (b) are necessary by Theorem 3.3.

To prove the sufficiency let $n_1 := \sum_{i=1}^{m} d(\theta_i)$ and $n_2 := \sum_{i=1}^{n} d(\alpha_i)$. Then

$$\alpha_i = \alpha'_i = 1, \quad i = n_2 + 1, \dots, n. \tag{4.2}$$

By Lemmas 4.2 and 2.1 we can assume that

$$(A, B) = \left( \begin{bmatrix} A_1 & A_3 \\ 0 & A_2 \end{bmatrix}, \begin{bmatrix} B_1 & Y \\ 0 & B_2 \end{bmatrix} \right),$$

where

$$(A_1, B_1) \in \mathbb{R}^{(n_1 - d(\theta_m))\times(n_1 - d(\theta_m))} \times \mathbb{R}^{(n_1 - d(\theta_m))\times(m-1)}$$

has $\theta_1, \dots, \theta_{m-1}$ as diagonal Hermite polynomials and

$$(A_2, B_2) \in \mathbb{R}^{(n_2 + d(\theta_m))\times(n_2 + d(\theta_m))} \times \mathbb{R}^{(n_2 + d(\theta_m))\times 1}$$

has $\theta_m$ as the diagonal Hermite polynomial and $\alpha_1, \dots, \alpha_{n_2 + d(\theta_m)}$ as invariant factors.

From (a) and (4.2), we have that

$$\alpha_1 \cdots \alpha_k \mid \alpha'_1 \cdots \alpha'_k \omega, \qquad k = 1, \dots, n_2 + d(\theta_m),$$
$$\alpha_1 \cdots \alpha_{n_2 + d(\theta_m)} = \alpha'_1 \cdots \alpha'_{n_2 + d(\theta_m)} \omega.$$

Let $\epsilon > 0$. By Corollary 4.8, there exists a matrix pair

$$(A'_2, B'_2) \in \mathbb{R}^{(n_2 + d(\theta_m))\times(n_2 + d(\theta_m))} \times \mathbb{R}^{(n_2 + d(\theta_m))\times 1}$$

such that $\| \, [A_2' \; B_2'] - [A_2 \; B_2] \, \| < \frac{\epsilon}{2}$, the invariant factors of $(A_2', B_2')$ are $\alpha_1', \ldots, \alpha_{n_2+d(\theta_m)}'$, and the diagonal Hermite polinomial of $(A_2', B_2')$ is $\theta_m' = \vartheta_m \omega$, where $r(\vartheta_m) = r(\theta_m)$ and $d(\vartheta_m) = d(\theta_m)$.

Let $(\overline{A}, \overline{B}) := \left( \begin{bmatrix} A_1 & A_3 \\ 0 & A_2' \end{bmatrix}, \begin{bmatrix} B_1 & Y \\ 0 & B_2' \end{bmatrix} \right)$. This matrix pair has $\alpha_1', \ldots, \alpha_n'$ as invariant factors and $\theta_1, \ldots, \theta_{m-1}, \theta_m'$ as diagonal Hermite polynomials.

Bearing in mind the Kalman decomposition we have

$$(\overline{A}, \overline{B}) \overset{s}{\sim} \left( \begin{bmatrix} \overline{A}_1 & \overline{A}_3 \\ 0 & \overline{A}_2 \end{bmatrix}, \begin{bmatrix} \overline{B}_1 \\ 0 \end{bmatrix} \right),$$

where $(\overline{A}_1, \overline{B}_1) \in \mathbb{R}^{(n_1+d(\omega)) \times (n_1+d(\omega))} \times \mathbb{R}^{(n_1+d(\omega)) \times m}$ is a controllable pair with $\theta_1, \ldots, \theta_{m-1}, \theta_m'$ as the diagonal Hermite polynomials and $\overline{A}_2 \in \mathbb{R}^{(n_2-d(\omega)) \times (n_2-d(\omega))}$ has the same nontrivial invariant factors as $(\overline{A}, \overline{B})$.

Since $d(\theta_m') = d(\theta_m \omega)$ and $r(\theta_m') = r(\theta_m \omega)$ and bearing in mind (b), it is easily seen that there exist monic polynomials $\beta_i' \in \mathbb{R}[s]$, $1 \le i \le m$, $d(\beta_i') = d(\beta_i)$, $1 \le i \le m$, such that

$$\beta_j' \cdots \beta_m' \mid \theta_j \cdots \theta_{m-1} \theta_m', \qquad j = 1, \ldots, m,$$

$$\beta_1' \cdots \beta_m' = \theta_1 \cdots \theta_{m-1} \theta_m'.$$

Now the theorem follows by applying Theorem 4.9 to $(\overline{A}_1, \overline{B}_1)$. $\square$

[illegible] 4.12. By using the results of [12] and [6], condition (a) of Theorem 4.11 can be written in terms of the Segre characteristic of the eigenvalues of $(A, B)$ as in Theorem 3.3.

## REFERENCES

[1] A. C. Antoulas, *New results on the algebraic theory of linear systems: The solution of the cover problems*, Linear Algebra Appl., 50 (1983), pp. 1–43.

[2] J. Barría and D. A. Herrero, *Closure of similarity orbits of nilpotent operators.* I. *Finite rank operators*, J. Operator Theory, 1 (1979), pp. 177–186.

[3] H. den Boer and G. Ph. A. Thijsse, *Semi-stability of sums of partial multiplicities under additive perturbation*, Integral Equations Operator Theory, 3 (1980), pp. 23–42.

[4] R. Gantmacher, *Matrix Theory*, Vol. I, Chelsea, New York, 1977.

[5] I. Gohberg and M. A. Kaashoek, *Unsolved problems in matrix and operator theory.* I. *Partial multiplicities and additive perturbations*, Integral Equations Operator Theory, 1 (1978), pp. 278–283.

[6] J. M. Gracia, I. De Hoyos, and I. Zaballa, *Perturbation of linear control systems*, Linear Algebra Appl., 121 (1989), pp. 353–383.

[7] U. Helmke, *Topology of the moduli space for reachable linear dynamical systems: The complex case*, Math. Systems Theory, 19 (1986), pp. 155–187.

[8] D. Hinrichsen and Prätzel-Wolters, *Generalized Hermite matrices and complete invariants of strict system equivalence*, SIAM J. Control Optim., 21 (1983), pp. 289–305.

[9] T. Kailath, *Linear Systems*, Prentice–Hall, Englewood Cliffs, NJ, 1980.

[10] P. Lancaster and M. Tismenestsky, *The Theory of Matrices with Applications*, Academic Press, London, 1985.

[11] C. C. Mac Duffee, *The Theory of Matrices*, Chelsea, New York, 1946.

[12] A. S. Markus and E. E. Parilis, *The change of the Jordan structure of a matrix under small perturbations*, Linear Algebra Appl., 54 (1983), pp. 139–152.

[13] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, 1979.

[14] D. Q. Mayne, *A canonical model for identification of multivariable systems*, IEEE Trans. Automat. Control, AC-17 (1972), pp. 728–729.

[15] V. M. Popov, *Invariant description of linear, time-invariant controllable systems*, SIAM J. Control Optim., 10 (1978), pp. 252–264.

[16] A. Pokrzywa, *On perturbations and the equivalence orbit of a matrix pencil*, Linear Algebra Appl., 82 (1986), pp. 99–121.

[17] H. H. Rosenbrock, *State-Space and Multivariable Theory*, Thomas Nelson and Sons, London, 1970.

[18] I. Zaballa, *Interlacing inequalities and control theory*, Linear Algebra Appl., 101 (1988), pp. 9–31.

[19] I. Zaballa, *Controllability and Hermite indices of matrix pairs*, Internat. J. Control, 68 (1997), pp. 61–86.

[20] I. Zaballa, *Asignación de invariantes mediante feedback*, lecture notes, Universidad Politécnica de Cataluña, Barcelona, 1991.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 473–495

# SEMICIRCLE LAW FOR HADAMARD PRODUCTS*

Z. D. BAI† AND L. X. ZHANG‡

**Abstract.** In this paper, assuming $p/n \to 0$ as $n \to \infty$, we will prove the weak and strong convergence to the semicircle law of the empirical spectral distribution of the Hadamard product of a normalized sample covariance matrix and a sparsing matrix, which is of the form $A_p = \frac{1}{\sqrt{np}}(X_{m,n}X_{m,n}^* - \sigma^2 n I_m) \circ D_m$, where the matrices $X_{m,n}$ and $D_m$ are independent and the entries of $X_{m,n}$ $(m \times n)$ are independent, the matrix $D_m$ $(m \times m)$ is Hermitian with independent entries above and on the diagonal, $p$ is the sum of the second moments of the row (and column) entries of $D_m$, and "$\circ$" denotes the Hadamard product of matrices.

**Key words.** dilute matrix, Hadamard product, large dimensional random matrix, Marčenko–Pastur law, random matrix theory, sample covariance matrix, semicircle law, sparse matrix, spectral distribution, Wigner matrix

**AMS subject classifications.** 15A52, 60F05, 62H99

**DOI.** 10.1137/050640424

## 1. Introduction.

The study of large random matrices started in the 1950s in the field of theoretical nuclear physics and from then on has attracted considerable interest of both theoretical physicists and statisticians. Initially, in theoretical nuclear physics, random matrices are constructed to model the interactions between a huge body of interacting atomic nuclei. Then the energy levels of these particles can be interpreted by the eigenvalues of the random matrices. The study of the eigenvalue behavior of large random matrices has been called the spectral theory of large random matrices.

Suppose $A_m$ is an $m \times m$ random matrix whose eigenvalues are denoted by $\{\lambda_j^{(m)} : 1 \le j \le m\}$. Then by putting suitable assumptions on $A_m$, such as $A_m$ being Hermitian, these eigenvalues are real numbers, and it is possible to define the following empirical distribution:

$$F^{A_m}(x) = \frac{1}{m}\#\{j : \lambda_j^{(m)} \le x\},$$

which is called the empirical spectral distribution (ESD) of $A_m$. Here #{...} denotes the number of elements contained in the set {...}. The spectral theory of large random matrices involves the limiting distribution of these ESDs. The first rigorous theoretical result in this field is considered to be the work of Wigner [25], in which it is proved that if $A_m$ is a Gaussian matrix, namely, $A_m$ is symmetric and its diagonal entries and those above the diagonal are independent normal variables with mean 0, and the variance of the diagonal elements is $2\sigma^2$ while that of the off-diagonal elements is $\sigma^2$, then the expected ESD, i.e., $EF^{\frac{1}{\sqrt{m}}A_m}$ of $\frac{1}{\sqrt{m}}A_m$, converges to the semicircle law with

---

*Received by the editors September 16, 2005; accepted for publication (in revised form) by R. Bhatia September 21, 2006; published electronically April 10, 2007.
http://www.siam.org/journals/simax/29-2/64042.html

†KLASMOE, School of Mathematics and Statistics, Northeast Normal University, Changchun, China 130024 (stabaizd@nenu.edu.cn). The work of this author was supported by NSFC grant 10571020 and NUS grant R-155-000-056-112.

‡Department of Statistics and Applied Probability, National University of Singapore, Singapore 117546 (scip1379@nus.edu.sg).

scale parameter $\sigma$ $F_{sc}(x)$, which is given by

$$\frac{d}{dx}F_{sc}(x) = \begin{cases} \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - x^2} & \text{if } |x| \le 2\sigma, \\ 0 & \text{otherwise.} \end{cases}$$

This theorem was later extended to the general Wigner matrix $A_m = [a_{ij}]$, which is a Hermitian matrix with independent complex entries above and on the diagonal satisfying the following:

(1) The means of all entries are 0 and the variance of the off-diagonal entries is $\sigma^2$ (the variance of the diagonal elements is allowed to be different from that of the off-diagonal entries).

(2) For any $\eta > 0$,

$$\frac{1}{m^2}\sum_{i,j=1}^{m} E|a_{ij}^2|I[|a_{ij}| > \eta\sqrt{m}] \to 0.$$

Another class of important random matrices are the sample covariance ones, which are defined as

$$S_n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i^*,$$

where $\mathbf{x}_i = (x_{1i}, \dots, x_{mi})'$ and $\{x_{ij}\}$ are i.i.d. with mean zero and variance $\sigma^2$, and $(.)^*$ denotes the complex conjugate of vectors or matrices. It is known that the ESD of $S_n$ a.s. tends to the Marcĕnko–Pastur law (see Marcĕnko and Pastur [17]) with density

$$\frac{d}{dx}F_y(x) = \begin{cases} \frac{1}{2\pi\sigma^2 xy}\sqrt{(b-x)(x-a)} & \text{if } a < x < b, \\ 0 & \text{otherwise,} \end{cases}$$

where $y = \lim m/n$ and $a = \sigma^2(1-\sqrt{y})^2$, $b = \sigma^2(1+\sqrt{y})^2$, with an additional point mass of $1 - 1/y$ at 0 when $y > 1$. Their law was later extended under the following condition:

$$\frac{1}{mn}\sum_{ij} E|x_{ij}^2|I[|x_{ij}| > \eta\sqrt{n}] \to 0.$$

Some properties of the extreme eigenvalues of Wigner matrices and sample covariance matrices can be found in Bai and Yin [5, 6] and Bai and Silverstein [2, 3]. For more references, the reader is referred to Bai [1].

These two types of matrices are asymptotically connected. Indeed, when $E|x_{11}|^4 < \infty$ and $m$ is fixed, the normalized sample covariance matrix $\sqrt{n}(S_n - \sigma^2 I_m)$ will tend to a Gaussian matrix. Therefore, it is conceivable that the ESD of $\sqrt{n/m}(S_n - \sigma^2 I_m)$ would tend to the semicircle law if $m \to \infty$ and $m/n \to 0$. This was confirmed in Bai and Yin [4].

In nuclear physics, since the particles move in a very high velocity in a small range, many exciting states in very short time cannot be observed. Generally, if a real physical system is not of full connectivity, the random matrix describing the interactions between the particles in the system will have a large proportion of zero elements. In this case, a sparse random matrix provides a more natural and relevant description of the system. Indeed, in neural network theory, the number of neurons in one person's brain is probably of several orders of magnitude larger than that of the dendrites connected with one individual neuron (see Grenander and Silverstein [12]). Sparse random matrices are adopted in modelling these partially connected systems in neural network theory. A sparse or dilute matrix is a random matrix in which some entries will be replaced by 0 if not observed. Sometimes a large portion of the entries of the interesting random matrix can be 0's. Due to its special application background, the sparse matrix has received special attention in quantum mechanics, atomic physics, neural networks, and many other areas. Some recent works on large sparse matrices and their applications to various areas can be found in, among others, [7, 9] for linear algebra, [8, 12] for neural networks, [10, 11, 13, 15, 16, 21, 23, 24] for algorithms and computing, [18] for finance modeling, [19, 26] for electrical engineering, [20] for bio-interactions, and [22] for theoretical physics.

A sparse matrix can be expressed by a Hadamard product. If $B_m = [b_{ij}]$ and $D_m = [d_{ij}]$ are two $m \times m$ matrices, then the Hadamard product $A_p = (A_{ij})$ with $A_{ij} = b_{ij} d_{ij}$ is denoted by

$$A_p = B_m \circ D_m.$$

The matrix $A_p$ is sparse if the elements $d_{ij}$ of $D_m$ take values 0 and 1 with $\sum_{i=1}^m P(d_{ij} = 1) = p = o(m)$. The index $p$ usually stands for the level of sparseness; i.e., after performing the Hadamard product, the resulting matrix will have $p$ nonzero elements per row on the average.

It is commonly assumed that the matrix $D_m$ is symmetric and its entries $\{d_{ij} : i \leq j\}$ are independent Bernoulli trials with $P(d_{ij} = 1) = p_{ij}$ and independent of the entries of the matrix $B_m$. In Kohrunzhy and Rodgers [14, 15], it is assumed that

$$p_{ij} = \frac{\alpha}{m^\beta},$$

with $0 \leq \beta \leq 1$, $0 < \frac{\alpha}{m^\beta} < 1$, and the entries of $B_m$ are centralized elements of a sample covariance matrix. More precisely, suppose $X_{m,n} = [x_{ij} : i = 1, 2, \ldots, m, j = 1, 2, \ldots, n]$ is an $m \times n$ matrix with independent entries of mean 0 and variance $\sigma^2$. Let the sample covariance matrix of $X_{m,n}$ be defined as $S_n = \frac{1}{n} X_{m,n} X_{m,n}^*$. Then, define $B_m = \sqrt{n/p}(S_n - \sigma^2 I_m)$ and $A_p = B_m \circ D_m$.

In the present paper, we shall consider a kind of Hadamard product of a normalized sample covariance matrix with a sparsing matrix (whose entries are not necessarily Bernoulli trials) and show the weak and strong convergence to the semicircle law of the ESD of this kind of Hadamard product. To this end, we make the following assumptions. In what follows, the entries of $D_m$ and $X_{m,n}$ are allowed to depend on $n$. For brevity, the dependence on $n$ is suppressed.

[illegible] $D_m$

(D1) $D_m = [d_{ij}]$ is $m \times m$ Hermitian with $\{d_{ij} : i \leq j\}$ independent complex random variables.

(D2) $\max_j \left| \sum_{i=1}^m p_{ij} - p \right| = o(p)$, where $p_{ij} = E|d_{ij}^2|$.

(D3.1) For some $\delta \in [0, 1/2]$, there exists a constant $C_1 > 0$ such that $\max_j \sum_i E|d_{ij}| \leq C_1 m^\delta p^{1-\delta}$.

(D3.2) For each $k > 2$ there is a constant $C_k$ such that

$$E|d_{ij}|^k \leq C_k p_{ij}. \tag{1.1}$$

We remark here that condition (D3.2) implies that $p_{ij}$ are uniformly bounded. In fact, $p_{ij} = E|d_{ij}|^2 \leq (E|d_{ij}|^4)^{1/2} \leq C_4$. Combining this fact with condition (D2), we indeed have $p \leq Km$, for some constant $K > 0$. In view of this relation between $p$ and $m$, we notice that if condition (D3.1) holds for some $\delta_0 \in [0, 1/2]$, then it must hold for every $\delta \geq \delta_0$, $\delta \in [0, 1/2]$. Therefore, we clarify here and in what follows that when we say condition (D3.1) holds for some $\delta^* \in [0, 1/2]$ we are referring to $\delta^*$ as the smallest value in $[0, 1/2]$ for which condition (D3.1) holds. In this sense, for any $0 \leq \delta_1 \leq \delta_2 \leq 1/2$, we say we have a stronger sparseness in the case when condition (D3.1) holds for $\delta_1$ than in the case when condition (D3.1) holds for $\delta_2$. Also note that in the case of the weakest sparseness, i.e., $\delta = 1/2$, condition (D3.1) is a direct consequence of Hölder's inequality and condition (D2); that is, no additional assumption is imposed on the first moments of the sparsing factors in this case.

[illegible] $X_{m,n}$

(X1) $X_{m,n} = [x_{ij}]$ is $m \times n$ consisting of independent random variables with $Ex_{ij} = 0$, $E|x_{ij}|^2 = \sigma^2$.

(X2.1) For any $\eta > 0$,

$$\frac{1}{mn}\sum_{ij} E|x_{ij}^2|I[|x_{ij}| > \eta\sqrt[4]{np}] \to 0.$$

(X2.2) For any $\eta > 0$,

$$\sum_{u=1}^{\infty} \frac{1}{mn}\sum_{ij} E|x_{ij}^2|I[|x_{ij}| > \eta\sqrt[4]{np}] < \infty,$$

where $u$ may take $[p]$, $m$, or $n$.

(X3) For any $\eta > 0$,

$$\frac{1}{m}\sum_{i=1}^{m} P\left(\left|\sum_{k=1}^{n}(|x_{ik}|^2 - \sigma^2)d_{ii}\right| > \eta\sqrt{np}\right) \to 0. \tag{1.2}$$

We shall prove the following theorem.

THEOREM 1.1. (i) [illegible] (1.1) [illegible] (1.2) [illegible]
(ii) [illegible] $D_m$ [illegible] $X_{m,n}$
(iii) $p/n \to 0$ [illegible] $p \to \infty$
(iv) [illegible] (D3.1) [illegible] $\delta = 1/2$ [illegible] $m/n \to 0$ [illegible] (D3.1) [illegible] $\delta \in (0, 1/2)$ [illegible] $m \leq Kn$ [illegible] $K$ [illegible] (D3.1) [illegible] $\delta = 0$ [illegible] $m$ [illegible] $n$

[illegible] $F^{A_p}$ [illegible] $\sigma^2$ [illegible] $[p] \to \infty$ [illegible] $A_p = \frac{1}{\sqrt{np}}(X_{m,n}X_{m,n}^* - \sigma^2 nI_m) \circ D_m$ [illegible] (X2.1) [illegible] $[p] \to \infty$ [illegible] $m \to \infty$ [illegible] (X2.2) [illegible] $u = [p]$ [illegible] $u = m$ [illegible]

[illegible] 1.1. Theorem 1.1 covers all well-known results on sparse matrices, since the Bernoulli trials satisfy conditions (D3.1) (with $\delta = 0$) and (D3.2) obviously. The new contribution of Theorem 1.1 is to allow the sparsing factors $d_{ij}$ to be very nonhomogenous. Consider the following example. Let $D_m = [d_{ij}]$ be symmetric. Let $m = kL$ with $L$ fixed, and let for all $(\ell - 1)k < i, j \leq \ell k$ with $\ell \leq L$,

$$P(d_{ij} = 1) = p/k = 1 - P(d_{ij} = 0),$$

and for all other indices $i, j$, $d_{ij} \equiv 0$. Then conditions (D1), (D2), (D3.1), and (D3.2) are true whenever $p \le k$.

1.2. In the case of the strongest sparseness, condition (D3.1) seems not to allow the $d_{ij}$'s to take large values. In fact, it is not the case. For example, consider that

$$d_{ij} = c_n^{-1}|z_{ij}|I(|z_{ij}| > c_n),$$

where $z_{ij}$ are i.i.d. $N(0, 1)$ subject to the condition $d_{ij} = d_{ji}$ and $c_n$ is a positive constant uniquely solving the equation $Ez_{ij}^2 I(|z_{ij}| > c_n) = c_n^2 p/m$. Then obviously $d_{ij}$ can take very large values, and $D_m$ is symmetric with

$$\sum_{i=1}^{m} p_{ij} = c_n^{-2} \sum_{i=1}^{m} Ez_{ij}^2 I(|z_{ij}| > c_n) = p;$$

i.e., conditions (D1) and (D2) are satisfied.

Now we show that condition (D3.1) holds for $\delta = 0$ if $p/m \to 0$. In fact, we can see that if $p/m \to 0$, then $c_n \to \infty$ and consequently

$$Ez_{ij}^2 I(|z_{ij}| > c_n) \simeq 2c_n \varphi(c_n),$$

which implies that

$$\frac{p}{m} \simeq \frac{2}{c_n}\varphi(c_n),$$

where the notation "$\simeq$" is used to represent the relation that the two quantities on its two sides have a ratio which tends to 1 as $n \to \infty$, while $\varphi(\cdot)$ is the density function of standard normal variables.

Therefore, we get

$$\sum_{i=1}^{m} E|d_{ij}| = mc_n^{-1} E|z_{ij}|I(|z_{ij}| > c_n) \simeq 2mc_n^{-1}\phi(c_n) \simeq p,$$

and

$$\begin{aligned} E|d_{ij}|^k &= c_n^{-k} E|z_{ij}|^k I(|z_{ij}| > c_n) \\ &\simeq 2c_n^{-k} c_n^{k-1} \phi(c_n) = 2c_n^{-1}\phi(c_n) \\ &\simeq p/m = E|d_{ij}|^2, \end{aligned}$$

which implies that condition (D3.1) holds for $\delta = 0$, and that condition (D3.2) holds.

1.3. However, if condition (D3.1) is assumed for $\delta = 0$, then sometimes it may happen that the $d_{ij}$'s are not allowed to take small values with large probabilities. For example,

$$d_{ij} = \sqrt{p/m} \text{ with probability 1.}$$

Then obviously (D3.1) holds for and only for $\delta = 1/2$, i.e., the weakest sparseness. For this case, condition (D3.1) holds automatically when condition (D3.2) is assumed. Condition (D3.2), assuming that higher moments of the $d_{ij}$'s are not larger than a multiple of their second moments, is not seriously restrictive because the $d_{ij}$'s are usually small random variables.

Nonetheless, if condition (D3.1) is assumed for $\delta = 1/2$, the condition does allow the first moments of $d_{ij}$'s to be much larger than their second moments, e.g., $d_{ij} = c\sqrt{p/m}|z_{ij}|$, where $z_{ij}$ are i.i.d. random variables subject to the restriction $d_{ij} = d_{ji}$ and $c$ makes $Ed_{ij}^2 = p/m$. As a price, however, we need to require $m$ to have a smaller order than that of $n$, that is, $m/n \to 0$. The requirement is necessary in some sense. In section 3, we will present a counterexample showing that the semicircle law fails to hold if this requirement is not satisfied.

[illegible] 1.4. Note that $p$ may not be an integer, and it may increase very slowly as $n$ increases. Thus, the limit for $p \to \infty$ may not be true for almost sure convergence. So, we consider the limit when the integer part of $p$ tends to infinity. However, if we consider the convergence in probability, Theorem 1.1 is true for $p \to \infty$.

[illegible] 1.5. Conditions (D2) and (D3.2) imply that $p \le Km$; that is, the order of $p$ cannot be larger than that of $m$. In the theorem, it is assumed that $p/n \to 0$; that is, $p$ also has a lower order than $n$. This is essential. However, the relation between $m$ and $n$ can be arbitrary if condition (D3.1) holds for $\delta = 0$. It is important to remind the readers that the statement "the relation between $m$ and $n$ can be arbitrary" only says that there are examples with $m/n \to \infty$ as well as examples with $m/n \to 0$, for which the results in Theorem 1.1 are applicable equally well once condition (D3.1) is satisfied with $\delta = 0$. For example, if the $d_{ij}$'s (subject to the condition $d_{ij} = d_{ji}$) are the Bernoulli trials defined by $P(d_{ij} = 1) = p/m = 1 - P(d_{ij} = 0)$ for any $i$, $j$, then $E|d_{ij}| = E|d_{ij}|^2 = E|d_{ij}|^k$ for any $k > 2$. This implies conditions (D1), (D2), (D3.1) (with $\delta = 0$), and (D3.2) always hold. But no matter $m/n \to 0$, $m/n \le K < \infty$, or $m/n \to \infty$, Theorem 1.1 always holds, provided that $p/n \to 0$.

[illegible] 1.6. From the proof given in the next sections, one can see that the almost sure convergence is true for $m \to \infty$ in all places except the part of the truncation on the entries of $X_{m,n}$ which was guaranteed by condition (X2.2). Thus, if condition (X2.2) holds for $u = m$, then the almost sure convergence is true in the sense of $m \to \infty$. Sometimes, it may be of interest to consider the almost sure convergence in the sense of $n \to \infty$. Examining the proof given in the next sections, one can find that to guarantee the almost sure convergence for $n \to \infty$, the removal of the diagonal elements of the matrix requires $m/\log n \to \infty$; the truncation on the entries of $X_{m,n}$ requires condition (X2.2) to be true for $u = n$. As for Proposition 3.1, as remarked in section 3, one may modify the conclusion of (II) on page 11 as

$$E|M_k - EM_k|^{2\mu} = O(m^{-\mu})$$

for any fixed integer $\mu$. Thus, if $m \ge n^{\varepsilon}$ for some positive constant $\varepsilon$, then the almost sure convergence of the ESD of the matrix after the truncation and centralization is true for $n \to \infty$. We therefore see that the conclusion of Theorem 1.1 can be strengthened to the almost sure convergence as $n \to \infty$ under the additional assumptions that, for some small positive constant $\varepsilon$, $m \ge n^{\varepsilon}$ and condition (X2.2) holds for $u = n$.

[illegible] 1.7. In Theorem 1.1, if $p = m$ and $d_{ij} \equiv 1$ for all $i$ and $j$ and the entries of $X_{m,n}$ are i.i.d., then the model considered in Theorem 1.1 reduces to that of Bai and Yin [4], where it is assumed that the fourth moment of $x_{ij}$ is finite. It can be easily verified that the conditions of Theorem 1.1 are satisfied under Bai and Yin's assumption. Thus, Theorem 1.1 contains Bai and Yin's result as a special case.

[illegible] 1.8. If there is a positive and increasing function $\varphi(x)$ defined on $\mathbb{R}^+$ such that

$$Q_n \equiv \frac{1}{mn}\sum_{ij} E|x_{ij}^2|\varphi(|x_{ij}|)I[|x_{ij}| > \eta\sqrt[4]{np}] \to 0, \tag{1.3}$$

then condition (X2.1) holds. Letting $\varphi(x) = x^{4(2\nu-1)}$ with $\frac{1}{2} \le \nu < 1$, (1.3) reduces to condition (2.4) of Kohrunzhy and Rodgers [14], if we change their notation to $p_{ij} = P(d_{ij} = 1) = p/m$ with $p = n^{2\nu-1}$ and $m/n \to c \in (0,\infty)$. It can be verified easily that all other assumptions of Theorem 1.1 are satisfied under their conditions; condition (X3) is automatically true since $P(d_{ii} \ne 0) = 0$ was assumed in their paper. Thus Theorem 1.1 covers Kohrunzhy and Rodgers [14] as a special case for all $\nu$'s in the interval $[1/2, 1)$.

Furthermore, it can be seen that if $Q_n \to 0$ with a suitable rate such that

$$\sum_{u=1}^{\infty} Q_n/\varphi(\eta\sqrt[4]{np}) < \infty, \tag{1.4}$$

then condition (X2.2) is satisfied. Indeed, $\varphi(x) = x^{4(2\nu-1)}$, with $(1+\sqrt{5})/4 < \nu < 1$, makes (1.4) hold. Then for these $\nu$'s Theorem 1.1 states the almost sure convergence holds and hence is stronger than the conclusion of i.p. convergence proven by Kohrunzhy and Rodgers [14].

[illegible] 1.9. The most important contribution of Theorem 1.1 to the random matrix theory is to allow nonhomogeneous and nonzero-one sparseness and for the case of the strongest sparseness to allow an arbitrary relation between $m$ and $n$. The conditions on the entries of $X_{m,n}$ are to require some homogeneity on the $X_{m,n}$ matrix. We conjecture that the homogeneity on the $X_{m,n}$ matrix can be relaxed if we require the entries of the $D_m$ matrix to have certain homogeneity. This problem is under investigation.

The organization of this paper is as follows. In section 2, we apply the truncation and centralization technique to the matrix $A_p$ so that the proof of Theorem 1.1 can proceed under more convenient conditions and hence can be simplified. Then in section 3, for the matrix resulting from the truncation and centralization treatment, we prove the convergence of the moments of their ESDs, and from the result we infer the validity of Theorem 1.1.

## 2. Truncation and centralization.

In this section, we apply the truncation and centralization techniques to $A_p$. However, we first need to present some preliminary results useful in subsequent proofs. These results consist of several inequalities concerning the difference between two ESDs.

Lemma 2.1 (difference inequality). [illegible] $n \times n$ [illegible] $A$ [illegible] $B$ [illegible]

$$L^3(F^A, F^B) \le \frac{1}{n} tr(A-B)^2,$$

[illegible] $L(\cdot,\cdot)$ [illegible] é [illegible] $F$ [illegible] $G$ [illegible]

$$L(F,G) = \inf\{\varepsilon > 0 : F(x-\varepsilon) - \varepsilon < G(x) < F(x+\varepsilon) + \varepsilon \quad \forall x \in \mathbb{R}\}.$$

Lemma 2.2 (rank inequality). [illegible] $A$ [illegible] $B$ [illegible] $n \times n$ [illegible]

$$\|F^A - F^B\| \le \frac{1}{n}\text{rank}(A-B),$$

[illegible] $\|\cdot\|$ [illegible] $f(\cdot)$ [illegible] $\|f\| = \sup_{x\in\mathbb{R}}|f(x)|$

LEMMA 2.3 (Bernstein's inequality). $\{X_n\}$ 0 $M$ $S_n = \sum_{i=1}^n X_i$ $\eta_n = ES_n^2$

$$P(|S_n| > \varepsilon) \le 2\exp\left\{-\frac{\varepsilon^2}{2(M\varepsilon + \eta_n)}\right\}.$$

The proofs for the first two lemmas can be found in Bai [1]. With the aid of these lemmas, we can now develop the following results on applying the truncation and centralization techniques.

**2.1. Removal of the diagonal elements of $\boldsymbol{A_p}$.** For any $\varepsilon > 0$, denote by $\widehat{A}_p$ the matrix obtained from $A_p$ by replacing its diagonal elements whose absolute values are greater than $\varepsilon$ by 0 and denote by $\widetilde{A}_p$ the matrix obtained from $A_p$ by replacing all its diagonal elements by 0.

PROPOSITION 2.4. 1.1

$$\|F^{\widehat{A}_p} - F^{A_p}\| \to 0, \ a.s.$$

$$L^3(F^{\widehat{A}_p}, F^{\widetilde{A}_p}) \le \varepsilon^2.$$

The second conclusion of the proposition is a trivial consequence of Lemma 2.1. As for the first conclusion, by the rank inequality,

$$\|F^{\widehat{A}_p} - F^{A_p}\| \le \frac{1}{m}\sum_{i=1}^m I\left[\left|\frac{1}{\sqrt{np}}\sum_{k=1}^n (|x_{ik}|^2 - \sigma^2)d_{ii}\right| > \varepsilon\right].$$

By condition (X3) in (1.2), we have

$$\sum_{i=1}^m P\left(\left|\frac{1}{\sqrt{np}}\sum_{k=1}^n (|x_{ik}|^2 - \sigma^2)d_{ii}\right| > \varepsilon\right) = o(m).$$

By Bernstein's inequality, it follows that for any constant $\eta > 0$,

$$\begin{aligned} P(\|F^{\widehat{A}_p} - F^{A_p}\| \ge \eta) &\le P\left(\sum_{i=1}^m I\left[\left|\frac{1}{\sqrt{np}}\sum_{k=1}^n (|x_{ik}|^2 - \sigma^2)d_{ii}\right| > \varepsilon\right] \ge \eta m\right) \\ &\le 2e^{-bm}, \end{aligned}$$

for some constant $b > 0$. By the Borel–Cantelli lemma, we conclude that

$$\|F^{\widehat{A}_p} - F^{A_p}\| \to 0, \ \text{a.s.}$$

Combining the two conclusions in Proposition 2.4, we have shown that

$$L(F^{A_p}, F^{\widetilde{A}_p}) \to 0, \ \text{a.s.}$$

Hence, in what follows, we can assume that the diagonal elements are 0; i.e., assume $d_{ii} = 0$ for all $i = 1, \ldots, m$.

**2.2. Truncation and centralization of the entries of $X_{m,n}$.** Note that condition (X2.1) in (1.2) guarantees the existence of $\eta_n \downarrow 0$ such that

$$\frac{1}{mn\eta_n^2}\sum_{ij} E|x_{ij}|^2 I(|x_{ij}| > \eta_n \sqrt[4]{np}) \to 0.$$

Similarly, if condition (X2.2) holds, there exists $\eta_n \downarrow 0$ such that

$$\sum_{u=1}^{\infty} \frac{1}{mn\eta_n^2}\sum_{ij} E|x_{ij}|^2 I[|x_{ij}| > \eta_n \sqrt[4]{np}] < \infty,$$

for $u$ takes $[p]$, $m$, or $n$. In the subsequent truncation procedure, we shall not distinguish under whichever condition the sequence $\{\eta_n\}$ is defined. The reader should remember that whatever condition is used, the $\{\eta_n\}$ is defined by that condition.

Define $\tilde{x}_{ij} = x_{ij}I[|x_{ij}| \le \eta_n \sqrt[4]{np}] - Ex_{ij}I[|x_{ij}| \le \eta_n \sqrt[4]{np}]$ and $\hat{x}_{ij} = x_{ij} - \tilde{x}_{ij}$. Also, define $\widetilde{B}_m$ with $\widetilde{B}_{ij} = \frac{1}{\sqrt{np}}\sum_{k=1}^{n} \tilde{x}_{ik}\bar{\tilde{x}}_{jk}$ $(i \neq j)$, and denote its Hadamard product with $D_m$ by $\widetilde{A}_p$. Then we have the following proposition.

Proposition 2.5. [illegible] (X2.1) [illegible] (1.2) [illegible] 1.1

$$L(F^{\widetilde{A}_p},\ F^{A_p}) \to 0,\ i.p.$$

[illegible] (X2.1) [illegible] (X2.2) [illegible]

$$L(F^{\widetilde{A}_p},\ F^{A_p}) \to 0,\ a.s. \text{ as }\ u \to \infty,$$

[illegible] $u = [p]$, $m$ [illegible] $n$ [illegible] $u$ [illegible] (X2.2) [illegible] By the difference inequality,

$$\begin{aligned} L^3(F^{\widetilde{A}_p},\ F^{A_p}) &\le \frac{1}{m}\mathrm{tr}[(B_m - \widetilde{B}_m)\circ D_m]^2 \\ &= \frac{1}{mnp}\sum_{i\neq j}\Big|\sum_{k=1}^{n}(x_{ik}\bar{x}_{jk} - \tilde{x}_{ik}\bar{\tilde{x}}_{jk})d_{ij}\Big|^2. \end{aligned}$$

We have

$$\begin{aligned} &E\left(\frac{1}{mnp}\sum_{i\neq j}\Big|\sum_{k=1}^{n}(x_{ik}\bar{x}_{jk} - \tilde{x}_{ik}\bar{\tilde{x}}_{jk})d_{ij}\Big|^2\right) \\ &= \frac{1}{mnp}\sum_{i\neq j}\sum_{k=1}^{n} E|x_{ik}\bar{x}_{jk} - \tilde{x}_{ik}\bar{\tilde{x}}_{jk}|^2 E|d_{ij}^2| \\ &\le \frac{8\sigma^2}{mnp}\sum_{j=1}^{m}\sum_{k=1}^{n} E|\hat{x}_{jk}|^2 \sum_{i=1}^{m} p_{ij} \\ &\le \frac{16\sigma^2}{mn}\sum_{j=1}^{m}\sum_{k=1}^{n} E|x_{jk}|^2 I[|x_{jk}| > \eta_n \sqrt[4]{np}], \end{aligned}$$

where in the last step we have used condition (D2), which can be easily shown to remain true after the removal of the diagonal elements of $D_m$. Thus we can see that if condition (X2.1) in (1.2) is assumed, then the right-hand side of the above inequality converges to 0 and hence the first conclusion follows.

However, if condition (X2.2) holds, then we have

$$\sum_{u=1}^{\infty}\frac{16\sigma^2}{mn}\sum_{ij}E|x_{ij}|^2 I(|x_{ij}|>\eta_n\sqrt[4]{np})<\infty,$$

and it follows that

$$L^3(F^{\widetilde{A}_p},\ F^{A_p})\to 0,\ \text{a.s.}$$

as $u\to\infty$, where $u$ takes $[p]$, $m$, or $n$ in accordance with the choice of $u$ in (X2.2). The proof of this proposition is complete.

From the above two propositions, we are allowed to make the following additional assumptions:

$$\text{(i) } d_{ii}=0.$$

$$\text{(ii) } Ex_{ij}=0,\quad |x_{ij}|\le\eta_n\sqrt[4]{np}. \tag{2.1}$$

Note that, we shall no longer have $E|x_{ij}|^2=\sigma^2$ after the truncation and centralization on the $X_{m,n}$ entries. Write $E|x_{ij}|^2=\sigma_{ij}^2$. We shall have the following proposition.

PROPOSITION 2.6. 1.1

(a) $\max_j|\sum_i E|d_{ij}|^2-p|=o(p)$,

(b) $i$ $j$ $\sigma_{ij}^2\le\sigma^2$ $\frac{1}{mn}\sum_{ij}\sigma_{ij}^2\to\sigma^2$

It is trivial to check the truth of (a) and the first part in (b). While the second part of (b) follows from condition (X2.1) and the following fact:

$$\begin{aligned}0&\le\sigma^2-\frac{1}{mn}\sum_{ij}\sigma_{ij}^2\\&\le\frac{2}{mn}\sum_{ij}E|x_{ij}|^2 I(|x_{ij}|>\eta_n\sqrt[4]{np})\to 0.\end{aligned}$$

**3. Proof of the theorem by the moment approach.** In the last section, we have shown that to prove Theorem 1.1, it suffices to do it under the additional conditions (i), (ii) in (2.1) and (a), (b) in Proposition 2.6. In the present section, we shall prove the following proposition.

PROPOSITION 3.1. 1.1 (i), (ii) (2.1) (a), (b) 2.6 $m\to\infty$ $F^{A_p}(x)$ $A_p$ $\sigma^2$

To prove the proposition, we employ the moment method. That is, based on the moment convergence theorem, by denoting $M_k$ and $m_k$ the $k$th moment of $F^{A_p}$ and the semicircle law $F_{sc,\sigma^2}(x)$ with scale parameter $\sigma^2$, we prove that $M_k\to m_k$ a.s. and that the sequence $\{m_k\}$ satisfies the Carleman condition $\sum_{k=1}^{\infty}m_{2k}^{-1/2k}=\infty$. It is easy to calculate that

$$m_k=\begin{cases}\frac{\sigma^{4s}(2s)!}{s!(s+1)!} & \text{if } k=2s,\\ 0 & \text{if } k=2s+1.\end{cases}$$

By noting that $m_{2k} \leq \sigma^{4k} 2^{2k}$, it is easy to see that the Carleman condition holds. Thus, to complete the proof of the proposition, by using the Borel–Cantelli lemma, we only need to prove

$$(\mathrm{I}) : E(M_k) = m_k + o(1),$$

and

$$(\mathrm{II}) : E|M_k - EM_k|^4 = O\left(\frac{1}{m^2}\right).$$

Now, we begin to proceed in the proof of (I) and (II) under conditions of Proposition 3.1. Write $\mathbf{i} = (i_1, \ldots, i_k)$, $\mathbf{j} = (j_1, \ldots, j_k)$, and by defining $i_{k+1} \equiv i_1$,

$$\begin{aligned} \mathcal{I} = \{(\mathbf{i}, \mathbf{j}) : 1 \leq i_v \leq m, 1 \leq j_v \leq n, \\ \text{with } i_v \neq i_{v+1} \text{ for each } 1 \leq v \leq k\}. \end{aligned} \tag{3.1}$$

Then, by definition we have

$$M_k = \frac{1}{mn^{k/2}p^{k/2}} \sum_{(\mathbf{i},\mathbf{j}) \in \mathcal{I}} d_{(\mathbf{i},\mathbf{j})} X_{(\mathbf{i},\mathbf{j})},$$

where

$$d_{(\mathbf{i},\mathbf{j})} = d_{i_1 i_2} \cdots d_{i_k i_1},$$

$$X_{(\mathbf{i},\mathbf{j})} = x_{i_1 j_1} \overline{x}_{i_2 j_1} x_{i_2 j_2} \overline{x}_{i_3 j_2} \cdots \overline{x}_{i_k j_{k-1}} x_{i_k j_k} \overline{x}_{i_1 j_k}.$$

For each pair $(\mathbf{i}, \mathbf{j}) = ((i_1, \ldots, i_k), (j_1, \ldots, j_k)) \in \mathcal{I}$, construct a graph $G(\mathbf{i}, \mathbf{j})$ by plotting the $i_v$'s and $j_v$'s on two parallel straight lines, respectively, and then drawing $k$ down edges $(i_v, j_v)$ from $i_v$ to $j_v$, $k$ up edges $(j_v, i_{v+1})$ from $j_v$ to $i_{v+1}$, and another $k$ horizontal edges $(i_v, i_{v+1})$ from $i_v$ to $i_{v+1}$. A down edge $(i_v, j_v)$ corresponds to the variable $x_{i_v j_v}$, an up edge $(j_v, i_{v+1})$ corresponds to the variable $\overline{x}_{i_{v+1} j_v}$, and a horizontal edge $(i_v, i_{v+1})$ corresponds to the variable $d_{i_v i_{v+1}}$. A graph corresponds to the product of the variables corresponding to the edges making up this graph. We shall call the subgraph of horizontal edges and their vertices of $G(\mathbf{i}, \mathbf{j})$ the roof of $G(\mathbf{i}, \mathbf{j})$ and denote it as $\overline{G}(\mathbf{i}, \mathbf{j})$, and call the subgraph of vertical edges and their vertices of $G(\mathbf{i}, \mathbf{j})$ the base of $G(\mathbf{i}, \mathbf{j})$ and denote it as $\underline{G}(\mathbf{i}, \mathbf{j})$. By noting that the roof of $G(\mathbf{i}, \mathbf{j})$ depends on $\mathbf{i}$ only, we may simplify the notation of roofs as $\overline{G}(\mathbf{i})$.

Two graphs $G(\mathbf{i}_1, \mathbf{j}_1)$ and $G(\mathbf{i}_2, \mathbf{j}_2)$ are said to be isomorphic if one can be converted to the other by a permutation on $(1, \ldots, m)$ and a permutation on $(1, \ldots, n)$. All graphs are classified into isomorphic classes. An isomorphic class is denoted by $\mathcal{G}$. Similarly, two roofs $\overline{G}(\mathbf{i}_1)$ and $\overline{G}(\mathbf{i}_2)$ are said to be isomorphic if one can be converted to the other by a permutation on $(1, \ldots, m)$. An isomorphic roof class is denoted by $\overline{\mathcal{G}}$. For a given $\mathbf{i}$, two graphs $G(\mathbf{i}, \mathbf{j}_1)$ and $G(\mathbf{i}, \mathbf{j}_2)$ are said to be isomorphic given $\mathbf{i}$ if one can be converted to the other by a permutation on $(1, \ldots, n)$. An isomorphic class given $\mathbf{i}$ is denoted by $\underline{\mathcal{G}}(\mathbf{i})$.

Then, we may rewrite

$$\begin{aligned} M_k &= \frac{1}{mn^{k/2}p^{k/2}} \sum_{\mathbf{i},\mathbf{j}} d_{\overline{G}(\mathbf{i})} X_{\underline{G}(\mathbf{i},\mathbf{j})} \\ &= \frac{1}{mn^{k/2}p^{k/2}} \sum_{\mathcal{G}} \sum_{G(\mathbf{i},\mathbf{j}) \in \mathcal{G}} d_{\overline{G}(\mathbf{i})} X_{\underline{G}(\mathbf{i},\mathbf{j})}. \end{aligned} \tag{3.2}$$

(I) 11 By the notation introduced above,

$$E(M_k) = \frac{1}{mn^{k/2}p^{k/2}} \sum_{\mathcal{G}} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} EX_{\underline{G}(\mathbf{i},\mathbf{j})}. \tag{3.3}$$

Note that when $G(\mathbf{i},\mathbf{j})$ contains a single vertical edge, $EX_{\underline{G}(\mathbf{i},\mathbf{j})} = 0$. Thus we may assume in the following that each graph appearing in the summation of $EM_k$ does not contain single vertical edges. Also note that, from the definition of the set $\mathcal{I}$, each graph also does not contain any loops of horizontal edges.

Let us denote by $l$, $r$, $s$ the numbers of noncoincident vertical edges, noncoincident $i_v$ vertices, and noncoincident $j_v$ vertices, respectively. It is obvious that these numbers must be the same for isomorphic graphs. Further, notice that for each isomorphic class $\mathcal{G}$, there is a unique graph $G(\mathbf{i},\mathbf{j})$ satisfying

$$i_1 = 1, i_{v+1} \le \max\{i_1, \ldots, i_v\} + 1,$$

$$j_1 = 1, j_{v+1} \le \max\{j_1, \ldots, j_v\} + 1,$$

which will be called the canonical (or representative) graph of $\mathcal{G}$. We can further define a number $q$ as follows. Suppose that $\mathcal{G}$ is an isomorphic class having the index $r$, the number of noncoincident $i$-vertices in its canonical graph $G(\mathbf{i},\mathbf{j})$. Then since $\overline{G}(\mathbf{i})$, the roof of $G(\mathbf{i},\mathbf{j})$, is a connected graph, we can select a tree which contains all the $r$ vertices of $\overline{G}(\mathbf{i})$ and exactly $r-1$ edges. Excluding the $r-1$ edges in $\overline{G}_1(\mathbf{i})$, we have $k-(r-1)$ edges left in $\overline{G}(\mathbf{i})$. Then the remaining $k-(r-1)$ edges together with all of the vertices of $\overline{G}(\mathbf{i})$ form another subgraph of $\overline{G}(\mathbf{i})$, which will be denoted by $\overline{G}_2(\mathbf{i})$. The remainder subgraph $\overline{G}_2(\mathbf{i})$ may not be connected. Suppose that $\overline{G}_2(\mathbf{i})$ consists of $q$ connected blocks with the understanding that each isolated vertex is considered as a connected block. Then by permutations of the $i$-vertices and $j$-vertices, we may use the same number $q$ for all the other graphs in this isomorphic class. Therefore, we may denote by $\mathcal{G}(l,r,s,q)$ the collection of isomorphic classes with the indices $l$, $r$, $s$, and $q$. We then have the following two propositions for estimating the terms involved in $EM_k$.

PROPOSITION 3.2. 3.1 $\mathcal{G} \in \mathcal{G}(l,r,s,q)$ $K$

$$\sum_{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}} E|d_{\overline{G}(\mathbf{i})}| \le Km^{1+\delta(q-1)}p^{r-1-\delta(q-1)} \tag{3.4}$$

$$\sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}} \\ I\,fixed}} E|d_{\overline{G}(\mathbf{i})}| \le Km^{\delta(q-1)}p^{r-1-\delta(q-1)}, \tag{3.5}$$

$I$ $i$ $\overline{G}(\mathbf{i})$. $K$ $I$

It is straightforward to see that (3.4) is a consequence of (3.5). We thus need only prove (3.5). Since $\overline{G}(\mathbf{i})$ is connected, there must exist $q-1$ edges in $\overline{G}_1(\mathbf{i})$ which make the $q$ blocks of $\overline{G}_2(\mathbf{i})$ connected. From the definition of $\overline{G}_1(\mathbf{i})$, these $q-1$ edges are single in $\overline{G}(\mathbf{i})$ and, together with their vertices, cannot form any cycles. We shall call them bridge edges. Suppose the $(q-1)$ bridge edges are

$$(i_{b_1}, i_{b_1+1}), (i_{b_2}, i_{b_2+1}), \ldots, (i_{b_{q-1}}, i_{b_{q-1}+1});$$

the other $r-q$ edges in $\overline{G}_1(\mathbf{i})$ are

$$(i_{a_1}, i_{a_1+1}), (i_{a_2}, i_{a_2+1}), \dots, (i_{a_{r-q}}, i_{a_{r-q}+1});$$

the $k-r+1$ edges in $\overline{G}_2(\mathbf{i})$ are

$$(i_{c_1}, i_{c_1+1}), (i_{c_2}, i_{c_2+1}), \dots, (i_{c_{k-r+1}}, i_{c_{k-r+1}+1}).$$

(Note that $q$ may be equal to 1 so that there are no bridge edges at all, but it is easy to see that the proof that follows is still valid.) Then by Hölder's inequality, we have

$$\begin{aligned}
&\sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} E|d_{\overline{G}(\mathbf{i})}| \\
&= E \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} |d_{i_{b_u} i_{b_u+1}}| \prod_{v=1}^{r-q} |d_{i_{a_v} i_{a_v+1}}| \prod_{w=1}^{k-r+1} |d_{i_{c_w} i_{c_w+1}}| \\
&\le \left( E \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} |d_{i_{b_u} i_{b_u+1}}| \prod_{v=1}^{r-q} |d_{i_{a_v} i_{a_v+1}}|^2 \right)^{1/2} \\
&\quad \left( E \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} |d_{i_{b_u} i_{b_u+1}}| \prod_{w=1}^{k-r+1} |d_{i_{c_w} i_{c_w+1}}|^2 \right)^{1/2} \\
&\le K m^{\delta(q-1)} p^{r-1-\delta(q-1)}.
\end{aligned}$$

To get this inequality, we used the following two results namely, by conditions (D2), (D3.1), and (D3.2),

$$\begin{aligned}
&E \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} |d_{i_{b_u} i_{b_u+1}}| \prod_{v=1}^{r-q} |d_{i_{a_v} i_{a_v+1}}|^2 \\
&= \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} E|d_{i_{b_u} i_{b_u+1}}| \prod_{v=1}^{r-q} E|d_{i_{a_v} i_{a_v+1}}|^2 \\
&\le K(m^\delta p^{1-\delta})^{q-1} p^{r-q} \\
&= K m^{\delta(q-1)} p^{r-1-\delta(q-1)},
\end{aligned}$$

and further by the fact that the $p_{ij}$'s are uniformly bounded,

$$\begin{aligned}
&E \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} |d_{i_{b_u} i_{b_u+1}}| \prod_{w=1}^{k-r+1} |d_{i_{c_w} i_{c_w+1}}|^2 \\
&= \sum_{\substack{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}\\ I fixed}} \prod_{u=1}^{q-1} E|d_{i_{b_u} i_{b_u+1}}| E \prod_{w=1}^{k-r+1} |d_{i_{c_w} i_{c_w+1}}|^2 \\
&\le K m^{\delta(q-1)} p^{r-1-\delta(q-1)}.
\end{aligned}$$

This completes the proof of the proposition.

Note that since each graph does not contain a single vertical edge, we have $l \leq k$. Since each graph does not contain any loops of horizontal edges, we further have $l \geq 2s$. And as a basic property of connected graph, we have $r + s \leq l + 1$. To estimate the integer $q$, we need the following proposition.

PROPOSITION 3.3. [illegible] $l$ $s$ $q$ [illegible]

$$l - 2s \geq q - 1. \tag{3.6}$$

[illegible] Since $\overline{G}(\mathbf{i})$ contains no loops, it follows that each $j$-vertex is connected with at least two noncoincident vertical edges and hence that $l - 2s \geq 0$, which implies (3.6) for the case of $q = 1$. Now assume $q > 1$. Then we have $q - 1$ bridge edges. If $(i_v, i_{v+1})$ is a bridge edge, then we call the vertex $j_v$ its supporting vertex, while the edges $(i_v, j_v)$, $(j_v, i_{v+1})$ are its supporting edges.

Denote the $s$ noncoincident $j$-vertices by $J_1, J_2, \ldots, J_s$. For each $1 \leq a \leq s$, denote by $l_a$ the number of noncoincident vertical edges connected with $J_a$. Then obviously $l = l_1 + l_2 + \cdots + l_s$. Note that each noncoincident $j$-vertex is composed of at least two $j$-vertices coincident with each other. For each $1 \leq a \leq s$, denote by $t_a$ the number of bridge edges supported by $J_a$. Here, of course, $t_a$ is the total number of bridge edges whose supporting $j$-vertex is from those coincident $j$-vertices constituting $J_a$. Then $q - 1 = t_1 + t_2 + \cdots + t_s$. To prove $l - 2s \geq q - 1$, it is sufficient to prove for each $1 \leq a \leq s$, $l_a \geq t_a + 2$.

If $t_a = 0$, then $l_a \geq t_a + 2$ follows simply from the previously stated fact that each $j$-vertex is connected with at least two noncoincident vertical edges. Now assume $t_a \geq 1$. In view of the property that bridge edges together with their vertices do not form any cycles and may be disconnected among themselves, we shall consider two cases, when the $t_a$ bridge edges (together with their vertices) form exactly one tree and when they form more than one tree disjoint with each other. For the first case, since each supporting edge connected with $J_a$ must take one vertex of the tree, there are exactly $t_a + 1$ noncoincident supporting edges connected with $J_a$. The same reasoning shows that for the second case, there are at least $t_a + 2$ noncoincident supporting edges connected with $J_a$, and hence $l_a \geq t_a + 2$.

To complete the proof of the proposition, we need only proceed with the proof of the first case. In this case, the tree formed by the $t_a$ bridge edges possesses $(t_a + 1)$ vertices. Arbitrarily select two vertices of the tree. Then these two vertices are joined by one path composed of only bridge edges from the tree. We assert that there cannot be any edge of $\overline{G}(\mathbf{i})$ which does not belong to the tree, taking the two vertices as its two endpoints. To see this, by the way of contradiction, we suppose that one such edge exists. Then this edge and the prescribed path form a cycle, and consequently we see that this edge must not belong to $\overline{G}_1(\mathbf{i})$ and that the cycle belongs to the graph consisting of $\overline{G}_2(\mathbf{i})$ and all bridge edges. Denote this later mentioned graph by $\overline{G}_3(\mathbf{i})$. It follows that removing any edge of the path from $\overline{G}_3(\mathbf{i})$ does not cause the graph to be disconnected, and so any edge arbitrarily selected from the bridge edges forming the path is not cutting in $\overline{G}_3(\mathbf{i})$. However, by the definition of bridge edges, $\overline{G}_3(\mathbf{i})$ is a connected graph and each of the $(q - 1)$ bridge edges should be cutting in the graph. Thus we reach a contradiction and conclude that our assertion is true.

Now arbitrarily select one vertex of degree one in the tree. Then the supporting edge connecting this vertex and $J_a$ must be single among supporting edges and must be coincident with one nonsupporting vertical edge. Note that this nonsupporting vertical edge may be a down edge and also may be an up edge. We first consider the case when this vertical edge is a down edge, say $(i_v, j_v)$. Then $i_v$ is coincident with the vertex we selected which has degree one in the tree, and $j_v$ is one of the coincident $j$-vertices constituting $J_a$. Note that since $(i_v, j_v)$ is nonsupporting, $(i_v, i_{v+1})$ is not a bridge edge. Recall that by (3.1), $i_v \neq i_{v+1}$. By the preceding argument, $i_{v+1}$ cannot be coincident with any of the other $t_a$ vertices of the tree either. This implies that the up vertical edge $(j_v, i_{v+1})$ cannot be coincident with any of the $(t_a + 1)$ noncoincident supporting edges connected with $J_a$. Therefore it follows that $l_a \geq t_a + 2$. For the other case when the vertical edge is an up edge, say $(j_v, i_{v+1})$, a similar argument can be used to conclude that the anterior down edge $(i_v, j_v)$ cannot be coincident with any of the $(t_a + 1)$ noncoincident supporting edges connected with $J_a$ (see Figure 3.1). The proof of the proposition is now complete.

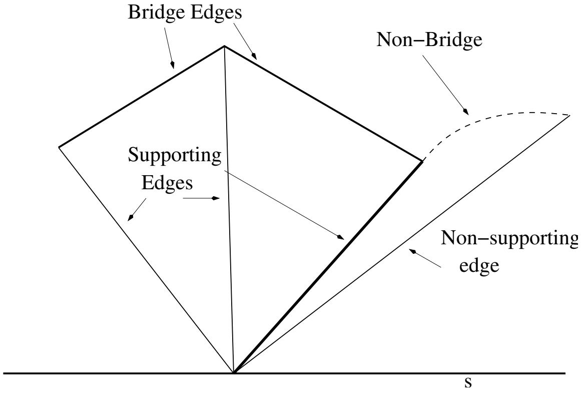


FIG. 3.1. *The definition of bridge edges.*

From Proposition 3.2, it follows for each $\mathcal{G} \in \mathcal{G}(l, r, s, q)$,

$$\begin{aligned}
&\frac{1}{mn^{k/2}p^{k/2}} \left| \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} EX_{\underline{G}(\mathbf{i},\mathbf{j})} \right| \\
&\leq \frac{1}{mn^{k/2}p^{k/2}} \sum_{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}} |Ed_{\overline{G}(\mathbf{i})}| \sum_{\underline{G}(\mathbf{i},\mathbf{j})\in\underline{\mathcal{G}}(\mathbf{i})} |EX_{\underline{G}(\mathbf{i},\mathbf{j})}| \\
&\leq \frac{1}{mn^{k/2}p^{k/2}} K m^{1+\delta(q-1)} p^{r-1-\delta(q-1)} n^{s} (\eta_n \sqrt[4]{np})^{2k-2l} \\
&= K\eta_n^{2(k-l)} m^{\delta(q-1)} n^{s-l/2} p^{r-l/2-1-\delta(q-1)} \\
&= K\eta_n^{2(k-l)} (m/n)^{\delta(q-1)} (p/n)^{\frac{l}{2}-s-\delta(q-1)} p^{r+s-l-1}. \qquad (3.7)
\end{aligned}$$

Based on this relation, we separate the terms involved in $EM_k$ into three parts; i.e., let $S_1$ be the sum of terms with $l < k$, $S_2$ be the sum of terms with $l = k$, but either $r + s < l + 1$ or $l > 2s$, and $S_3$ be the sum of terms with $l = k$, $r + s = l + 1$, $l = 2s$, and hence from Proposition 3.3, $q = 1$. Noting the relations between the numbers $l$, $r$, $s$ analyzed previously, we have

$$EM_k = S_1 + S_2 + S_3.$$

We first prove $S_1 \to 0$ and $S_2 \to 0$. Note that under the assumptions of Theorem 1.1, $(m/n)^{\delta(q-1)}$ is bounded. From Proposition 3.3, $l/2 - s \geq \delta(q-1)$ always holds.

Thus we get

$$|S_1| = o((m/n)^{\delta(q-1)}(p/n)^{\frac{l}{2}-s-\delta(q-1)}p^{r+s-l-1}) = o(1).$$

If $\delta(q-1) = 0$, then

$$|S_2| = O((p/n)^{\frac{k}{2}-s}p^{r+s-k-1}) = o(1),$$

since either $k > 2s$ or $r + s < k + 1$. If $\delta \in (0, 1/2)$ and $q > 1$, then

$$|S_2| = O((p/n)^{\frac{k}{2}-s-\delta(q-1)}) = o(1),$$

since $k/2 - s > \delta(q-1)$. If $\delta = 1/2$ and $q > 1$, then since $m/n \to 0$,

$$|S_2| = O((m/n)^{\frac{1}{2}(q-1)}) = o(1).$$

Note that when $k$ is odd, no terms involved in the summation of $EM_k$ belong to $S_3$ and hence we must have

$$EM_k \to 0.$$

In the following we only need to evaluate $S_3$ for when the case $k$ is even, by definition of $S_3$, $k = 2s$.

We first note that $r+s = k+1$ implies that there cannot be cycles of noncoincident vertical edges in the base of the graph. Also note that $l = k$ implies that each noncoincident vertical edge must consist of exactly two vertical edges. It follows that each down edge must coincide with one and only one up edge because the coincidence of a down edge (an up edge) with another down edge (up edge) would imply that the noncoincident edges of the base contain a cycle. Therefore, if we denote the noncoincident vertical edges by $\{(u_1, v_1), \ldots, (u_k, v_k)\}$, then

$$EX_{\underline{G}(\mathbf{i},\mathbf{j})} = \prod_{j=1}^{k} \sigma^2_{u_j v_j},$$

and hence for each isomorphic class $\mathcal{G} \in \mathcal{G}(2s, s+1, s, 1)$ (all isomorphic classes involved in $S_3$ constitute $\mathcal{G}(2s, s+1, s, 1)$), we have

$$\sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} EX_{\underline{G}(\mathbf{i},\mathbf{j})} = \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} \prod_{j=1}^{k} \sigma^2_{u_j v_j}.$$

Now we show that

$$\frac{1}{m(np)^s} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} \prod_{j=1}^{k} \sigma^2_{u_j v_j} = \frac{1}{m(np)^s} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} \sigma^{2k} + o(1). \tag{3.8}$$

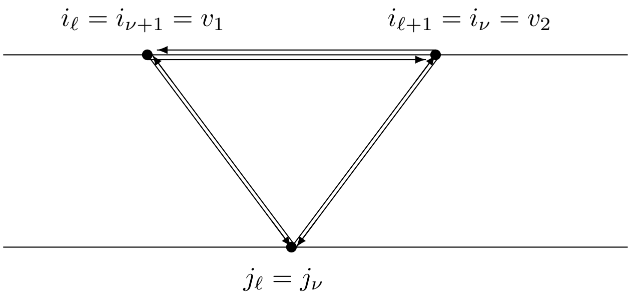


FIG. 3.2. $(v_1, v_2)$ *coincides with* $(v_2, v_1)$.

By (b) of Proposition 2.6 and (3.5) of Proposition 3.2 for the case $q = 1$, we have

$$\begin{aligned}
0 &\le \frac{1}{m(np)^s} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} |Ed_{\overline{G}(\mathbf{i})}| \left[\sigma^{2k} - \prod_{j=1}^{k} \sigma^2_{u_j v_j}\right] \\
&\le \frac{1}{m(np)^s} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} |Ed_{\overline{G}(\mathbf{i})}| \sum_{\ell=1}^{k} \left[\sigma^{2(k-\ell)}(\sigma^2 - \sigma^2_{u_\ell v_\ell}) \prod_{j=1}^{\ell-1} \sigma^2_{u_j v_j}\right] \\
&\le \frac{1}{m(np)^s} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} |Ed_{\overline{G}(\mathbf{i})}| \sum_{\ell=1}^{k} [\sigma^{2(k-1)}(\sigma^2 - \sigma^2_{u_\ell v_\ell})] \\
&\le \frac{\sigma^{2(k-1)}}{mnp^s} \sum_{\overline{G}(\mathbf{i})\in\overline{\mathcal{G}}} |Ed_{\overline{G}(\mathbf{i})}| \sum_{\ell=1}^{k} \sum_{v_\ell} (\sigma^2 - \sigma^2_{u_\ell v_\ell}) \\
&\le \sum_{\ell=1}^{k} \frac{K\sigma^{2(k-1)}}{mn} \sum_{v_\ell} \sum_{u_\ell} (\sigma^2 - \sigma^2_{u_\ell v_\ell}) \to 0,
\end{aligned}$$

from which (3.8) follows.

For a graph corresponding to a term in $S_3$, we claim that each horizontal edge $(v_1, v_2)$ must coincide with a horizontal edge $(v_2, v_1)$. In fact, suppose that $(i_\ell, i_{\ell+1})$ is the first appearance of $(v_1, v_2)$; i.e., $i_\ell = v_1$, $i_{\ell+1} = v_2$ and $v_2$ is not in $\{i_1, \ldots, i_\ell\}$. We claim that $j_\ell$ is not in $\{j_1, \ldots, j_{\ell-1}\}$. Otherwise, assuming $j_\ell$ is coincident with $j_a$ with $a < \ell$, then because of (3.1) and the definition of $i_{\ell+1}$, $i_a$, $i_{a+1}$, and $i_{\ell+1}$ are three noncoincident $i$-vertices so that $(i_a, j_a)$, $(j_a, i_{a+1})$, and $(j_\ell, i_{\ell+1})$ are three noncoincident vertical edges. It follows that there are at least three noncoincident vertical edges connected with the noncoincident $j$-vertex which contains $j_a$, $j_\ell$. This obviously violates the assumption $k = 2s$. As a consequence of the assertion, both of the two vertical edges $(i_\ell, j_\ell)$ and $(j_\ell, i_{\ell+1})$ are single up to the vertex $i_{\ell+1}$. In the future development of the graph, there must be one down edge $(i_\nu, j_\nu)$ coincident with the single upedge $(j_\ell, i_{\ell+1})$; that is, $i_\nu = i_{\ell+1} = v_2$ and $j_\nu = j_\ell$. Then the next up edge $(j_\nu, i_{\nu+1})$ must coincide with $(i_\ell, j_\ell)$ since, otherwise, the vertex $j_\nu = j_\ell$ will be connected with at least three noncoincident vertical edges. Thus $i_{\nu+1} = i_\ell = v_1$, and so the horizontal edge $(i_\ell, i_{\ell+1}) = (v_1, v_2)$ coincides with the horizontal edge

$(i_\nu, i_{\nu+1}) = (v_2, v_1)$ (see Figure 3.2). In view of the total number of noncoincident $i$-vertices contained in $\overline{G}(\mathbf{i})$ is $r = s+1$, we conclude that the noncoincident horizontal edges of $\overline{G}(\mathbf{i})$ form a tree of $s$ edges, each edge consisting of exactly two horizontal edges of converse directions.

It follows that

$$Ed_{\overline{G}(\mathbf{i})} = \prod_{\ell=1}^{s} p_{a_\ell, b_\ell},$$

where $(a_\ell, b_\ell)$, $1 \le \ell \le s$, denote the edges of the tree of noncoincident horizontal edges. By (3.8) and condition (D2), we have

$$\begin{aligned}
&\frac{1}{m(np)^s} \sum_{G(\mathbf{i},\mathbf{j}) \in \mathcal{G}} Ed_{\overline{G}(\mathbf{i})} EX_{\underline{G}(\mathbf{i},\mathbf{j})} \\
&= \frac{\sigma^{2k}}{mp^s} \sum_{\overline{G}(\mathbf{i}) \in \overline{\mathcal{G}}} \prod_{\ell=1}^{s} p_{a_\ell, b_\ell} + o(1) \\
&= \sigma^{2k} + o(1).
\end{aligned}$$

Therefore, to evaluate $EM_k$, what remains is to count the number of isomorphic classes in $\mathcal{G}(2s, s+1, s, 1)$. Note that for the graphs defined earlier, one only needs to arrange the vertical edges, since the positions of the horizontal edges will then be automatically determined by the positions of the $i$-vertices. When one draws the graph edge by edge, starting from $i_1$, an edge is called an innovation if it is the first appearance of a noncoincident edge and called a Type 3 edge otherwise. As we have shown in the previous paragraph, for a graph corresponding to a term in $S_3$, a down innovation must be followed by an up innovation, and a down edge of Type 3 must be followed by an up edge of Type 3. Thus, we only need to arrange the $s$ down innovations and the $s$ down edges of Type 3. Define $a_\ell = 1$ if the $\ell$th down edge is an innovation and $= -1$ otherwise. Before any Type 3 edge, there must be a single innovation. That is, for every $\ell \le k$, we should have

$$a_1 + \cdots + a_\ell \ge 0.$$

Thus, the number of isomorphic classes in $\mathcal{G}(2s, s+1, s, 1)$ is the number of sequences of $s$ ones and $s$ minus ones subject to the nonnegative partial sum requirement. By the reflection theorem, it is easy to show the number of such sequences is

$$\binom{2s}{s} - \binom{2s}{s-1} = \frac{(2s)!}{s!(s+1)!}.$$

Conclusion (I) on page 11 is proved.

[illegible] (II) [illegible] 11 Since $M_k$ is real, we consider

$$\begin{aligned}
&E(M_k - EM_k)^4 \\
&= \frac{1}{m^4 n^{2k} p^{2k}} \sum_{\substack{\mathbf{i}_\ell, \mathbf{j}_\ell \\ 1 \le \ell \le 4}} E\left( \prod_{\ell=1}^{4} [d_{\overline{G}(\mathbf{i}_\ell)} X_{\underline{G}(\mathbf{i}_\ell, \mathbf{j}_\ell)} - Ed_{\overline{G}(\mathbf{i}_\ell)} EX_{\underline{G}(\mathbf{i}_\ell, \mathbf{j}_\ell)}] \right),
\end{aligned}$$

where $G(\mathbf{i}_\ell, \mathbf{j}_\ell)$ is the graph defined by $(\mathbf{i}_\ell, \mathbf{j}_\ell)$ in the way given in the proof of (I) on page 11.

If $G(\mathbf{i}_\ell, \mathbf{j}_\ell)$ has no edges coincident with edges of the other three, then the corresponding term in the summation is 0 by independence. Furthermore, the term is also 0 if $\bigcup_{\ell=1}^4 G(\mathbf{i}_\ell, \mathbf{j}_\ell)$ contains a single vertical edge. Hence we need to consider only the following two cases:

(1) The four graphs are connected together through edges.

(2) $\bigcup_{\ell=1}^4 G(\mathbf{i}_\ell, \mathbf{j}_\ell)$ consists of two separated pieces, each of which is composed of two graphs connected together.

Split $E|M_k - EM_k|^4 = S_I + S_{II}$ according to the two cases. Denote the collection of graphs in case (1) by $\mathcal{C}_1$ and similar to the notation $\mathcal{C}_2$.

In case (1), the graph $G = \bigcup_{\ell=1}^4 G(\mathbf{i}_\ell, \mathbf{j}_\ell)$ has a connected roof. Let $r$, $s$, and $l$ be, respectively, the numbers of noncoincident $i$-vertices, noncoincident $j$-vertices, and noncoincident vertical edges contained in $G$. Similarly, we can define the number $q$. Then, similar to the estimation of $EM_k$, under the assumptions of Proposition 3.1, we have

$$
\begin{aligned}
|S_I| &\leq \frac{16}{m^4 n^{2k} p^{2k}} \sum_{G \in \mathcal{C}_1} E\left(\prod_{\ell=1}^4 |d_{\overline{G}(\mathbf{i}_\ell)}||X_{\underline{G}(\mathbf{i}_\ell, \mathbf{j}_\ell)}|\right) \\
&\leq \frac{K}{m^4 n^{2k} p^{2k}} \sum_{r,s,l,q} m^{1+\delta(q-1)} p^{r-1-\delta(q-1)} (\eta_n \sqrt[4]{np})^{(8k-2l)} n^s \\
&\leq K m^{-3} \sum_{r,s,l,q} \eta_n^{8k-2l} (m/n)^{\delta(q-1)} (p/n)^{\frac{l}{2}-s-\delta(q-1)} p^{r+s-l-1} \\
&= O(m^{-3}),
\end{aligned}
$$

where we have used Proposition 3.2 and the facts $l \leq 4k$, $r+s \leq l+1$, and $l-2s \geq q-1$.

To estimate $S_{II}$, one only needs to note that for each piece of the roof subgraph, there is one factor $m$ obtained and so totally there is one more factor $m$ obtained. Thus under the assumptions of Proposition 3.1, one gets that

$$S_{II} = O(m^{-2}).$$

Combining the above gives (II) on page 11. Consequently, we have completed the proof of Proposition 3.1 and Theorem 1.1.

[illegible] 3.1. Using the same approach as we prove (II) on page 11, one can easily show that

$$E|M_k - EM_k|^{2\mu} = O(m^{-\mu}), \tag{3.9}$$

for any fixed integer $\mu$. This result will be useful when the almost sure convergence is considered for $n \to \infty$.

At the end, we present two examples. The first example is to show, when condition (D3.1) is assumed for $\delta = 1/2$, to ensure the convergence of the semicircle law of $F^{A_p}$, it is necessary to require $m/n \to 0$.

[illegible] 3.1. Let $D_m = [d_{ij}]$ consist of $d_{ii} = 0$ and $d_{ij} = \sqrt{p/m}$ for $i \neq j$. Let $X_{m,n} = [x_{ij}]$ consist of i.i.d. standard normal random variables. Now assume $m/n \to c > 0$ and $p/n \to 0$. Then conditions (D2), (D3.1), and (D3.2) hold. Specifically, 1/2 is the smallest parameter in $[0, 1/2]$ such that condition (D3.1) is satisfied by $A_p$.

Consider the $k$th moment of $F^{A_p}$. Using the definitions we gave in proving Proposition 3.1, for any isomorphic class $\mathcal{G}$ whose canonical graph possesses $r$ noncoincident $i$-vertices and $s$ noncoincident $j$-vertices and does not contain loops of horizontal edges, we have

$$\begin{aligned} S_{\mathcal{G}} &= \frac{1}{mn^{k/2}p^{k/2}} \sum_{G(\mathbf{i},\mathbf{j})\in\mathcal{G}} Ed_{\overline{G}(\mathbf{i})} EX_{\underline{G}(\mathbf{i},\mathbf{j})} \\ &= K_{\mathcal{G}} m^{-1} n^{-k/2} p^{-k/2} (p/m)^{k/2} m^r n^s + o(1) \\ &= K_{\mathcal{G}} m^{r+s-k-1} (n/m)^{s-k/2} + o(1), \end{aligned}$$

where $K_{\mathcal{G}} = EX_{\underline{G}(\mathbf{i},\mathbf{j})}$. It is easy to see that

$$S_{\mathcal{G}} \to \begin{cases} 0 & \text{if } r+s < k+1, \\ K_{\mathcal{G}} c^{k/2-s} & \text{if } r+s = k+1. \end{cases}$$

Note that when $r+s = k+1$, since $r+s \le l+1$, where $l$ is the number of noncoincident vertical edges contained in the canonical graph of $\mathcal{G}$, it follows that $l \ge k$. If $l > k$, then there must exist a single vertical edge and hence $K_{\mathcal{G}} = 0$. Otherwise, $l = k$; then every noncoincident vertical edge is composed of exactly two vertical edges of opposite directions and hence $K_{\mathcal{G}} = 1$. Therefore, noticing the restriction that, since there are no loops of horizontal edges, every noncoincident $j$-vertex must be connected with at least two noncoincident vertical edges, we get

$$EM_k \to m_k = \sum_{s=1}^{[\frac{k}{2}]} c^{k/2-s} \mu_s,$$

where $\mu_s$ is the number of isomorphic classes whose canonical graphs satisfy the following condition:

(1) Each canonical graph contains exactly $s$ noncoincident $j$-vertices and $(k+1-s)$ noncoincident $i$-vertices.

(2) Each canonical graph contains exactly $k$ noncoincident vertical edges, each of which consists of two edges, of opposite directions.

(3) Each canonical graph possesses the property that, supposing one person starts a walk along its edges, then whenever a down edge leads to a new noncoincident $j$-vertex the next up edge must lead to a new $i$-vertex.

To estimate the limit $m_k$, let us observe further that

$$\begin{aligned} &\left(\frac{m}{n}\right)^{k/2} \times EM_k \\ &= m^{-1} n^{-k} \sum_{i_1,\dots,i_k}^{res} \sum_{j_1,\dots,j_k} E(x_{i_1 j_1} x_{i_2 j_1} x_{i_2 j_2} x_{i_3,j_2} \cdots x_{i_k j_k} x_{i_1 j_k}), \quad (3.10) \\ &\le E m^{-1} tr \left(\frac{1}{n} X_{m,n} X_{m,n}^*\right)^k, \quad (3.11) \end{aligned}$$

where the summation $\sum_{i_1,\dots,i_k}^{res}$ is taken over all possible values of $i_1,\dots,i_k$ satisfying the restriction that $i_1 \ne i_2$, $i_2 \ne i_3, \dots, i_k \ne i_1$. Thus it follows, by Theorem 2.5 of Bai [1], that $c^{k/2} m_k$ is bounded by the $k$th moment of the Marcĕnko–Pastur law with ratio index $c$ and scale index 1. Thus $\{m_k\}_{k=1}^{\infty}$ satisfies the Carleman condition.[1]

---

[1] Note that there is a one-to-one correspondence between $\mathcal{G}$ and its base and that the base of $\mathcal{G}$ must be a canonical graph defined in deriving the Marcĕnko–Pastur law. Thus, we indeed have

$$\mu_s \le \frac{1}{k+1-s}\binom{k}{s}\binom{k-1}{s-1}.$$

Using (3.10), one can easily show that

$$E(M_k - EM_k)^{2\mu} = O(m^{-2\mu}).$$

Therefore, with probability one, $F^{A_p}$ converges to a nonrandom limiting distribution, say $F$. It is easy to verify that when $k = 3$, we have $s = 1$ and $r = 3$ so that $i_1 \neq i_2 \neq i_3$ and $j_1 = j_2 = j_3$; i.e., there is exactly one contributing isomorphic class $\mathcal{G}$. Thus,

$$m_3 = \sqrt{c}.$$

Since the third moment of $F$ is not 0, $F$ is not the semicircle law. That is, we have shown with probability one that $F^{A_p}$ converges weakly but the limiting spectral distribution is not the semicircle law.

The next example is to show for the case when condition (D3.1) is assumed for $\delta \in (0, 1/2)$ that the condition $m/n$ is bounded and also necessary for the convergence to the semicircle law.

Example 3.2. Let $D_m = [d_{ij}]$ be defined as in Example 3.1. We assume the same conditions $m/n \to c > 0$ and $p/n \to 0$. Now we define $\tilde{D}_h = D_m \otimes I_h$ and $\tilde{B}_h = \frac{1}{\sqrt{np}}(X_{mh,n}X^*_{mh,n} - \sigma^2 n I_{mh})$, where "$\otimes$" denotes the Kronecker product of matrices, $h = [m^\eta]$ with $\eta > 0$, and $X_{mh,n}$ is $mh \times n$ consisting of i.i.d. standard normal random variables.

Let $\tilde{A}_p = \tilde{B}_h \circ \tilde{D}_h$. Then $\tilde{A}_p = \operatorname{diag}[A_{1,m}, \ldots, A_{h,m}]$, where

$$A_{i,m} = B_{ii} \circ D_m,\ i = 1, \ldots, h,$$

and $B_{ii}$ is the $i$th $m \times m$ major submatrix of $\tilde{B}_h$.

Note that $A_{1,m}, \ldots, A_{h,m}$ are independent with the same distribution as $A_p$ defined in Example 3.1. Denote by $\tilde{M}_k$, $M_{i,k}$, and $M_k$, respectively, the $k$th moment of $\tilde{A}_p$, the $k$th moment of $A_{i,m}$, and the $k$th moment of $A_p$. Then it follows that $EM_{i,k} = EM_k$ and $E(M_{i,k} - EM_{i,k})^{2\mu} = E(M_k - EM_k)^{2\mu}$. Since $F^{\tilde{A}_p} = \frac{1}{h}\sum_{i=1}^h F^{A_{i,m}}$ so that $\tilde{M}_k = \frac{1}{h}\sum_{i=1}^h M_{i,k}$, we get $E\tilde{M}_k = EM_k$ and $E(\tilde{M}_k - E\tilde{M}_k)^{2\mu} \le E(M_k - EM_k)^{2\mu}$. By the results we proved in Example 3.1, it follows with probability one that $F^{\tilde{A}_p}$ converges weakly but the limiting spectral distribution is not the semicircle law.

Let us now check the validity of the assumptions of Theorem 1.1 for $\tilde{A}_p$. Conditions (D1), (D2), and (D3.2) hold for $\tilde{A}_p$ automatically by definition. We now show that for any $\delta \in (0, 1/2)$ by choosing $\eta > 0$ such that $2\delta(1 + \eta) = 1$, condition (D3.1) is satisfied by $\tilde{A}_p$ for the given $\delta$. To see this, note that the dimension of $\tilde{A}_p$ is $mh$, and so we have

$$\sum_i Ed_{ij} \le \sqrt{mp} \le \frac{\sqrt{m}}{(mh)^\delta}(mh)^\delta p^{1-\delta} \le C_1 (mh)^\delta p^{1-\delta}.$$

By requiring $p = O(\log m)$, we can further see for any $\delta_0 < \delta$ that

$$\left(\sum_i Ed_{ij}\right) / \left((mh)^{\delta_0} p^{1-\delta_0}\right) \ge \frac{1}{2} m^{\frac{1}{2}(1-\delta_0/\delta)} p^{\delta_0 - \frac{1}{2}} \to \infty,$$

which confirms that $\delta$ is the smallest parameter in $(0, 1/2)$ such that condition (D3.1) is satisfied by $\tilde{A}_p$. Noticing that $mh/n \to \infty$, we see that $\tilde{A}_p$ satisfies all assumptions of Theorem 1.1 except only the condition that in case of $\delta \in (0, 1/2)$ the ratio between the vector dimension and the sample size should be bounded. We achieved our target.

## REFERENCES

[1] Z. D. Bai, *Methodologies in spectral analysis of large dimensional random matrices, A review*, Statist. Sinica, 9 (1999), pp. 611–677.

[2] Z. D. Bai and J. W. Silverstein, *No eigenvalues outside the support of the limiting spectral distribution of large dimensional sample covariance matrices*, Ann. Probab., 26 (1998), pp. 316–345.

[3] Z. D. Bai and J. W. Silverstein, *Exact separation of eigenvalues of large-dimensional sample covariance matrices*, Ann. Probab., 27 (1999), pp. 1536–1555.

[4] Z. D. Bai and Y. Q. Yin, *Convergence to the semicircle law*, Ann. Probab., 16 (1988), pp. 863–875.

[5] Z. D. Bai and Y. Q. Yin, *Necessary and sufficient conditions for almost sure convergence of the largest eigenvalue of a Wigner matrix*, Ann. Probab., 16 (1988), pp. 1729–1741.

[6] Z. D. Bai and Y. Q. Yin, *Limit of the smallest eigenvalue of a large dimensional sample covariance matrix*, Ann. Probab., 21 (1993), pp. 1275–1294.

[7] R. P. Barry and R. K. Pace, *Monte Carlo estimates of the log determinant of large sparse matrices*, in Linear Algebra and Statistics, Istanbul, 1997, Linear Algebra Appl. 289, North–Holland, New York, 1999, pp. 41–54.

[8] M. P. Bekakos and A. A. Bartzi, *Sparse matrix and network minimization schemes*, in Computational Methods and Neural Networks, Dynamic, Atlanta, GA, 1999, pp. 61–94.

[9] D. Boley and T. Goehring, *LQ-Schur projection on large sparse matrix equations*, in Preconditioning Techniques for Large Sparse Matrix Problems in Industrial Applications, Minneapolis, MN, 1999, Numer. Linear Algebra Appl. 7, John Wiley and Sons, Chichester, UK, 2000, pp. 491–503.

[10] E. F. F. Botta and F. W. Wubs, *Matrix renumbering ILU: An effective algebraic multilevel ILU preconditioner for sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1007–1026.

[11] G. M. Del Corso and F. Romani, *Heuristic spectral techniques for the reduction of bandwidth and work-bound of sparse matrices*, in memory of W. Gross., Numer. Algorithms, 28 (2001), pp. 117–136.

[12] U. Grenander and J. W. Silverstein, *Spectral analysis of networks with random topologies*, SIAM J. Appl. Math., 32 (1977), pp. 499–519.

[13] A. Gupta, *Improved symbolic and numerical factorization algorithms for unsymmetric sparse matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 529–552.

[14] A. Khorunzhy and G. J. Rodgers, *Eigenvalue distribution of large dilute random matrices*, J. Math. Phys., 38 (1997), pp. 3300–3320.

[15] A. Khorunzhy and G. J. Rodgers, *On the Wigner law in dilute random matrices*, Rep. Math. Phys., 42 (1998), pp. 297–319.

[16] Y. Lin, *Graph extensions and some optimization problems in sparse matrix computations*, Adv. Math. (China), 30 (2001), pp. 9–21.

[17] V. A. Marčenko and L. A. Pastur, *Distribution for some sets of random matrices*, Math. USSR-Sb., 1 (1967), pp. 457–483.

[18] R. Marti, M. Laguna, F. Glover, and V. Campos, *Reducing the bandwidth of a sparse matrix with tabu search, Financial modelling*, European J. Oper. Res., 135 (2001), pp. 450–459.

[19] B. J. McKenzie and T. Bell, *Compression of sparse matrices by blocked Rice coding*, IEEE Trans. Inform. Theory, 47 (2001), pp. 1223–1230.

[20] J.-M. Naulin, *A contribution of sparse matrices tools to matrix population model analysis*, in Deterministic and Stochastic Modeling of Biointeraction, West Lafayette, IN, 2000, Math. Biosci., 177/178 (2002), pp. 25–38.

[21] N. Neuss, *A new sparse-matrix storage method for adaptively solving large systems of reaction-diffusion-transport equations*, Computing, 68 (2002), pp. 19–36.

[22] D. A. Stariolo, E. M. F. Curado, and F. A. Tamarit, *Distributions of eigenvalues of ensembles of asymmetrically diluted Hopfield matrices*, J. Phys. A., 29 (1996), pp. 4733–4739.

[23] P. S. Vassilevski, *Sparse matrix element topology with application to AMG(e) and preconditioning*, in Preconditioned Robust Iterative Solution Methods, PRISM '01 (Nijmegen), Numer. Linear Algebra Appl. 9, John Wiley and Sons, Chichester, UK, 2002, pp. 429–444.

[24] K. Wang and J. Zhang, *MSP: A class of parallel multistep successive sparse approximate*

*inverse preconditioning strategies*, SIAM J. Sci. Comput., 24 (2003), pp. 1141–1156.

[25] E. P. Wigner, *On the distributions of the roots of certain symmetric matrices*, Ann. of Math. (2), 67 (1958), pp. 325–327.

[26] M. Y. Xia, C. H. Chan, S. Q. Li, B. Zhang, and L. Tsang, *An efficient algorithm for electromagnetic scattering from rough surfaces using a single integral equation and multilevel sparse-matrix canonical-grid method*, IEEE Trans. Antennas and Propagation, 51 (2003), pp. 1142–1149.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 496–529

# LOW RANK PERTURBATION OF KRONECKER STRUCTURES WITHOUT FULL RANK*

FERNANDO DE TERÁN[†] AND FROILÁN M. DOPICO[†]

**Abstract.** Let $P(\lambda) = A_0 + \lambda A_1$ be a singular $m \times n$ matrix pencil without full rank whose Kronecker canonical form (KCF) is given. Let $\rho$ be a positive integer such that $\rho \leq \min\{m, n\} - \operatorname{rank}(P)$ and $\rho \leq \operatorname{rank}(P)$. We study the change of the KCF of $P(\lambda)$ due to perturbation pencils $Q(\lambda)$ with $\operatorname{rank}(Q) = \rho$. We focus on the generic behavior of the KCF of $(P + Q)(\lambda)$, i.e., the behavior appearing for perturbations $Q(\lambda)$ in a dense open subset of the pencils with rank $\rho$. The most remarkable generic properties of the KCF of the perturbed pencil $(P + Q)(\lambda)$ are (i) if $\lambda_0$ is an eigenvalue of $P(\lambda)$, finite or infinite, then $\lambda_0$ is an eigenvalue of $(P + Q)(\lambda)$; (ii) if $\lambda_0$ is an eigenvalue of $P(\lambda)$, then the number of Jordan blocks associated with $\lambda_0$ in the KCF of $(P + Q)(\lambda)$ is equal to or greater than the number of Jordan blocks associated with $\lambda_0$ in the KCF of $P(\lambda)$; (iii) if $\lambda_0$ is an eigenvalue of $P(\lambda)$, then the dimensions of the Jordan blocks associated with $\lambda_0$ in $(P + Q)(\lambda)$ are equal to or greater than the dimensions of the Jordan blocks associated with $\lambda_0$ in $P(\lambda)$; (iv) the row (column) minimal indices of $(P + Q)(\lambda)$ are equal to or greater than the largest row (column) minimal indices of $P(\lambda)$. Moreover, if the sum of the row (column) minimal indices of the perturbations $Q(\lambda)$ is known, apart from their rank, then the whole set of the row (column) minimal indices of $(P + Q)(\lambda)$ is generically obtained, and in the case $\rho < \min\{m, n\} - \operatorname{rank}(P)$ the whole KCF of $(P + Q)(\lambda)$ is generically determined.

**Key words.** Kronecker canonical form, low rank perturbations, matrix spectral perturbation theory, mosaic Toeplitz matrices

**AMS subject classifications.** 15A21, 15A22, 15A18, 65F15

**DOI.** 10.1137/060659922

## 1. Introduction.

Matrix spectral canonical forms are very important both in theory and in applications like the behavior of dynamical systems near bifurcations. Spectral canonical forms are mathematical structures that are very fragile under perturbations. For instance, it is well known that, although the *Jordan* canonical form of a matrix $A$ has blocks of dimension larger than one, all the blocks in the Jordan form of the perturbed matrix $A + E$ have dimension one and correspond to eigenvalues different from those of $A$, for almost all perturbations $E$. The same can be said on the behavior of the *Weierstrass* canonical form of a regular matrix pencil $A_0 + \lambda A_1$, and on the *Kronecker* canonical form (KCF) of singular matrix pencils. In this latter case, in fact, the perturbed pencil has full rank for almost all perturbations. However, there are perturbations that allow us to guarantee that some part of the spectral canonical form of the original pencil is also a part of the spectral canonical form of the perturbed pencil. One example of perturbations of this kind is *low rank perturbations*, i.e., perturbations with a fixed rank that is small in some way specified by a property of the unperturbed matrix or pencil.

Low rank perturbations of spectral canonical forms have received attention since the 1980s. At least two kinds of contributions can be considered in this area. Given an

---

*Received by the editors May 15, 2006; accepted for publication (in revised form) by I. C. F. Ipsen October 19, 2006; published electronically April 10, 2007. This research was partially supported by the Ministerio de Educación y Ciencia of Spain through grants BFM-2003-00223 and MTM2006-06671, and by the PRICIT Program of the Comunidad de Madrid through the SIMUMAT Project (Ref. S-0505/ESP/0158).

http://www.siam.org/journals/simax/29-2/65992.html

[†] Departamento de Matemáticas, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain (fteran@math.uc3m.es, dopico@math.uc3m.es).

$m \times n$ pencil (or matrix) $P(\lambda)$ and perturbations $Q(\lambda)$ with fixed rank, the first class of works tries to classify all the spectral canonical forms of $(P+Q)(\lambda)$ compatible with the canonical form of $P(\lambda)$ and the rank of the perturbations $Q(\lambda)$. As far as we know, this has only been done for rank one perturbations; see [1] and [18] in this context. A second class of papers in this area characterizes generic properties of the spectral canonical form of $(P+Q)(\lambda)$, i.e., properties that hold for perturbations in a dense open subset of the matrices or pencils with a fixed rank; for this problem, see the references [4, 10, 13, 16, 17]. Generic properties have been considered only for the Jordan canonical form of matrices and for the Weierstrass canonical form of regular matrix pencils, and the study of explicit necessary and sufficient conditions for the generic behaviors to hold has been performed only in [4, 13]. The purpose of this paper is to determine generic properties of the KCF of singular matrix pencils without full rank under certain low rank perturbations, and to provide sufficient conditions for these properties to hold.

Throughout this work the term *generic* will frequently be used. This word appears in many mathematical works, but it is not a well-defined technical term, and its precise meaning is not always the same in the literature. In this paper, we use *generic* in the following sense: *[illegible] $\mathcal{C}$ [illegible] $\mathcal{C}$.* In our context, $\mathcal{C}$ will be the set of allowable perturbations, and we identify the set of $m \times n$ complex matrix pencils, $A_0 + \lambda A_1$, with $\mathbb{C}^{2mn}$, where the usual topology is considered. Therefore every subset $\mathcal{C}$ of pencils can be seen as a subset of $\mathbb{C}^{2mn}$. In this setting, we have that a set $\mathcal{G} \subset \mathcal{C}$ is dense in $\mathcal{C}$ if and only if every element in $\mathcal{C}$ is the limit of a sequence of elements in $\mathcal{G}$, and we will say that $\mathcal{G}$ is open in $\mathcal{C}$ if $\mathcal{G}$ is the intersection of $\mathcal{C}$ with an open subset of $\mathbb{C}^{2mn}$; i.e., we consider in $\mathcal{C}$ the *relative topology* induced by the usual topology of $\mathbb{C}^{2mn}$. To finish these comments on the term *generic*, let us remark that it will not be used in the statement of most theorems, where precise assumptions will be included. Discussions on the genericity of these assumptions will be separately addressed.

We will consider as unperturbed pencil a singular $m \times n$ matrix pencil $P(\lambda)$ without full rank, i.e., $\operatorname{rank}(P) < \min\{m, n\}$. Given an integer number $\rho$ such that

$$0 < \rho \leq \min\{m, n\} - \operatorname{rank}(P), \tag{1}$$

and $\rho \leq \operatorname{rank}(P)$, the set of perturbations is restricted to pencils $Q(\lambda)$ with $\operatorname{rank}(Q) = \rho$. Notice that (1) and $\rho \leq \operatorname{rank}(P)$ are both low rank conditions imposed on the perturbations. Here, the rank has to be understood as the rank of matrix polynomials, which is also known as the *normal rank* of a pencil.

For the set of perturbations defined in the previous paragraph the first problem we deal with is to get information on the generic regular part of the perturbed pencil $(P+Q)(\lambda)$. This is addressed in section 4, where it is proved that, generically, if $\lambda_0$ is an eigenvalue of $P(\lambda)$, finite or infinite, then $\lambda_0$ is also an eigenvalue of $(P+Q)(\lambda)$ with partial multiplicities greater than or equal to the corresponding partial multiplicities of $\lambda_0$ relative to $P(\lambda)$. These results are consequences of Theorem 4.4, *[illegible]*. The second problem we deal with is to get information on the generic minimal indices of $(P+Q)(\lambda)$. For the sake of brevity, let us summarize the results only for the column or right minimal indices. Similar results hold for the row minimal indices. It is known that the number of column minimal indices of $P(\lambda)$ is $n - \operatorname{rank}(P)$. The initial result we present is that, generically, $P + Q$ has $n - \operatorname{rank}(P) - \rho$ column minimal indices. This implies, in particular, that if $\rho = n - \operatorname{rank}(P)$, then $P + Q$ has no column minimal indices; i.e., it has full column rank. These results follow from Theorem 3.1 and its direct consequence, Corollary 3.2. The case $\rho < n - \operatorname{rank}(P)$ is much more difficult, and it is addressed in Theorem 5.8, where all the column minimal indices of $P + Q$ are generically determined if, apart from the rank, the sum of the column minimal indices of the perturbations $Q(\lambda)$ is known. As a corollary, Theorem 5.10 presents generic partial information on the column minimal indices of $P + Q$ when $\rho = \operatorname{rank}(Q)$ is the only property known on the perturbations. Loosely speaking, one can say that the generic column minimal indices of $P + Q$ are equal to or greater than the $n - \operatorname{rank}(P) - \rho$ largest column minimal indices of $P$. Theorems 5.8 and 5.10 constitute [illegible]. All the results previously described remain valid in the limit case

$$\rho = \min\{m, n\} - \operatorname{rank}(P).$$

If the strict inequality is assumed in (1), i.e., $\operatorname{rank}(P) + \operatorname{rank}(Q) < \min\{m, n\}$, it is possible to fully determine the generic KCF of $(P + Q)(\lambda)$ in terms of the sums of the column and row minimal indices and of the regular part of the KCF of $Q(\lambda)$. In the case that $\operatorname{rank}(Q)$ is the only information available on the perturbations, the generic KCF of $(P + Q)(\lambda)$ can only be partially determined. These results appear in Theorems 6.2 and 6.3, which are [illegible]. It should be stressed that all the generic results on the KCF of $(P + Q)(\lambda)$ that we present are very easy to describe, although to prove that they occur under certain generic sufficient conditions is a hard task that requires techniques very different from those used in [4, 13].

The class of low rank perturbations considered in this work includes very interesting problems. To cite one of them: the study of the generic variation of the minimal indices of a square pencil ($m = n$) under low rank perturbations requires necessarily the assumptions $\operatorname{rank}(P) < n$ (because otherwise $P(\lambda)$ has no minimal indices) and $\operatorname{rank}(P) + \operatorname{rank}(Q) < n$ (because otherwise generically $\operatorname{rank}(P + Q) = n$ and $P + Q$ has no minimal indices). However, this class of perturbations does not cover all the relevant situations. There are still open problems in the area of generic low rank perturbations of spectral canonical forms. Some of them will be discussed in section 7, where we will explain why the results obtained in this paper are, apart from being relevant by themselves, an essential step towards the solution of new open problems.

The perturbations considered in this work are not of small norm. The change of KCF of matrix pencils under small normwise perturbations was studied in [14], where the set of Kronecker structures nearby to a given one was characterized in terms of some majorization conditions on the sequences of column and row minimal indices and on the regular structure. Further results of this kind were obtained in [2] and [5].

Low rank perturbations of spectral properties have appeared in several applied problems. For instance, in the area of structural modifications of dynamical systems, it is of particular relevance to study how a system must be modified in order to fix certain eigenvalues in the new system. This is known generically as the "pole-zero assignment" problem [15]. In [7], low rank perturbations of the damping matrices of vibrating systems are considered in order to obtain defective systems.

The paper is organized as follows. In section 2 the notation and some preliminary results are introduced. In section 3 the meaning and genericity of the low rank assumptions used in different sections of this work are discussed, and, as a consequence, the generic number of row and column minimal indices of the perturbed pencil is determined. In section 4 generic properties of the regular structure of the perturbed pencil $(P + Q)(\lambda)$ are established. Section 5 deals with the minimal indices of $(P + Q)(\lambda)$. Section 6 describes the whole generic KCF of $(P + Q)(\lambda)$, assuming that the strict inequality $\rho < \min\{m, n\} - \operatorname{rank}(P)$ holds. Finally, in section 7 the conclusions and some open profblems are presented.

**2. Notation, definitions, and preliminary results.** Several basic definitions and results are presented in this section. Some of them are well known and are stated just to establish the notation used throughout the paper. In addition, some other definitions and elementary results are presented.

**2.1. Kronecker canonical form and rank of a pencil.** We begin by introducing the concepts of singular pencil, rank or normal rank of a pencil, and eigenvalue of a pencil.

DEFINITION 2.1 (see [8, Chapter XII]). [illegible] $A_0, A_1 \in \mathbb{C}^{m\times n}$ [illegible] $m\times n$ [illegible] matrix pencil

$$P(\lambda) = A_0 + \lambda A_1 \tag{2}$$

[illegible] singular [illegible] $m \neq n$ [illegible] $m = n$ [illegible] $\det(P(\lambda))$ [illegible] $\lambda$ [illegible] regular

DEFINITION 2.2. [illegible] $P(\lambda)$ [illegible] $\lambda$ [illegible] $P(\lambda)$ [illegible] $\operatorname{rank}(P)$ [illegible] $\lambda$

The [illegible] of a pencil is also called its [illegible] [2, 5]. However, we prefer the classical name [illegible], because this concept corresponds to the usual rank of matrices whose entries are rational functions of $\lambda$.

DEFINITION 2.3. [illegible] $\mu$ [illegible] $P(\lambda)$ [illegible] $P(\mu)$ [illegible] $\operatorname{rank}(P)$ [illegible] $P(\lambda) = A_0 + \lambda A_1$ [illegible] infinite [illegible] dual pencil $A_1 + \lambda A_0$

For every pencil $P(\lambda)$ there exist two nonsingular matrices $R \in \mathbb{C}^{m\times m}$ and $S \in \mathbb{C}^{n\times n}$ such that $R\,P(\lambda)\,S = \mathcal{K}_P(\lambda)$ is the KCF of $P(\lambda)$ (see [8, Chapter XII]). The KCF is a block diagonal matrix and is unique up to permutations of the diagonal blocks. To be more precise,

$$\mathcal{K}_P(\lambda) = \operatorname{diag}(L_{\varepsilon_1}, \ldots, L_{\varepsilon_p}, L^T_{\eta_1}, \ldots, L^T_{\eta_q}, \mathcal{J}_P), \tag{3}$$

where $L_{\varepsilon_i}$ is the $\varepsilon_i \times (\varepsilon_i + 1)$ matrix pencil

$$L_{\varepsilon_i} = \begin{bmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \end{bmatrix},$$

the superscript $T$ means transposition, and $\mathcal{J}_P$ is a square pencil that constitutes the [illegible] of the KCF of $P(\lambda)$. The matrix pencil $\mathcal{J}_P$ contains the spectral information on the eigenvalues of $P(\lambda)$. This means that $\mathcal{J}_P$ is a direct sum of Jordan blocks

$$J_k(\lambda_i) = \begin{bmatrix} \lambda-\lambda_i & 1 & & \\ & \lambda-\lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda-\lambda_i \end{bmatrix}_{k\times k},$$

associated with certain finite eigenvalues $\lambda_i \in \mathbb{C}$ of $P(\lambda)$, and, eventually, of Jordan blocks associated with the infinite eigenvalue

$$J_k(\infty) = \begin{bmatrix} 1 & \lambda & & \\ & 1 & \ddots & \\ & & \ddots & \lambda \\ & & & 1 \end{bmatrix}_{k\times k}.$$

The numbers $\varepsilon_1, \ldots, \varepsilon_p$ are called the *column minimal indices of* $P(\lambda)$, and $\eta_1, \ldots, \eta_q$ are called the *row minimal indices of* $P(\lambda)$ [8, Chapter XII]. Notice that the row minimal indices of $P(\lambda)$ are the column minimal indices of $P(\lambda)^T$ and vice versa. We will assume that they are indexed in nondecreasing order, i.e.,

$$0 \le \varepsilon_1 \le \varepsilon_2 \le \cdots \le \varepsilon_p \quad \text{and} \quad 0 \le \eta_1 \le \eta_2 \le \cdots \le \eta_q.$$

Analogously the matrix pencils $L_{\varepsilon_i}$ ($L_{\eta_j}^T$) are called the column or right (row or left) singular blocks of the KCF of $P(\lambda)$. These blocks reveal the *singular structure* of $P(\lambda)$.

Observe that, if the KCF of $P(\lambda)$ is given by (3), then

$$\operatorname{rank}(P) = n - p = m - q; \tag{4}$$

i.e., the rank of a pencil is related to the number of column and row singular blocks in its KCF. Notice also that if $\mathcal{J}_P$ is a $j \times j$ pencil, then

$$\operatorname{rank}(P) = j + \varepsilon_1 + \cdots + \varepsilon_p + \eta_1 + \cdots + \eta_q. \tag{5}$$

**2.2. The vector space of $n$-tuples of rational functions. Minimal bases.** The entries of an $m \times n$ pencil $P(\lambda) = A_0 + \lambda A_1$ are polynomials of degree one over $\mathbb{C}$. Moreover, it is well known that the column (row) minimal indices of $P(\lambda)$ are related to the degrees of certain polynomial solutions of $(A_0 + \lambda A_1)x(\lambda) = 0$ ($(A_0 + \lambda A_1)^T y(\lambda) = 0$) [8, Chapter XII], where $x(\lambda)$ ($y(\lambda)$) is an $n$-tuple ($m$-tuple) whose entries are polynomials. The vector $x(\lambda)$ will be called a *vector polynomial*. Previous comments make clear that vector polynomials can naturally arise in dealing with singular pencils. The set of polynomials with complex coefficients is a *ring* but not a *field*. This means that to extend many elementary ideas of linear algebra to vector polynomials one has to consider the field of all rational functions with complex coefficients. For instance, let $v_1 = [1+\lambda, 1+\lambda]^T$ and $v_2 = [1+\lambda^2, 1+\lambda^2]^T$ be two vector polynomials. The determinant of the matrix $[v_1|v_2]$ is obviously zero, but a rational function has to be necessarily used as a coefficient to express $v_2$ as a linear combination of $v_1$: $v_2 = \frac{1+\lambda^2}{1+\lambda} v_1$. The *field of rational functions* with complex coefficients will be denoted by $\mathbb{C}(\lambda)$, and the *vector space* over $\mathbb{C}(\lambda)$ *of* $n$-*tuples of rational functions* will be denoted by $\mathbb{C}^n(\lambda)$.

The following definitions are taken from [6] (see also [11]). The *degree*, $\deg(x)$, *of a vector polynomial* $x(\lambda)$ is the greatest degree of its components. Every vector subspace $\mathcal{V}$ of $\mathbb{C}^n(\lambda)$ always has a basis consisting of vector polynomials. It can be obtained from a general basis simply by multiplying each vector by the denominators of its entries. The *order* of such a polynomial basis is defined as the sum of the degrees of its vectors. A *minimal basis* of $\mathcal{V}$ is a polynomial basis of $\mathcal{V}$ that has least order among all polynomial bases of $\mathcal{V}$.

Let us introduce some additional concepts that we will use very often. Given an $m \times n$ matrix pencil $P(\lambda) = A_0 + \lambda A_1$, the [illegible] $P(\lambda)$ is the subspace of $\mathbb{C}^n(\lambda)$ ($\mathbb{C}^m(\lambda)$), $\mathcal{N}(P) = \{x(\lambda) \in \mathbb{C}^n(\lambda) : P(\lambda)x(\lambda) = 0\}$ ($\mathcal{N}(P^T) = \{y(\lambda) \in \mathbb{C}^m(\lambda) : P^T(\lambda)y(\lambda) = 0\}$). A [illegible] of $P(\lambda)$ is a [illegible] contained in $\mathcal{N}(P)$ ($\mathcal{N}(P^T)$). A [illegible] $P(\lambda)$ [illegible] is a minimal basis, $\{x_1(\lambda), \dots, x_p(\lambda)\}$, of $\mathcal{N}(P)$ with $\deg(x_1) \leq \deg(x_2) \leq \cdots \leq \deg(x_p)$. A [illegible] $P(\lambda)$ [illegible] is a minimal basis, $\{y_1(\lambda), \dots, y_q(\lambda)\}$, of $\mathcal{N}(P^T)$ with $\deg(y_1) \leq \deg(y_2) \leq \cdots \leq \deg(y_q)$.

Lemma 2.4 shows that the degrees of the vectors in an ROMB (LOMB) of $P(\lambda)$ are equal to the column (row) minimal indices of $P(\lambda)$.

Lemma 2.4. [illegible] $\varepsilon_1 \leq \cdots \leq \varepsilon_p$ [illegible] $\eta_1 \leq \cdots \leq \eta_q$ [illegible] $P(\lambda)$ [illegible] $\{x_1(\lambda), \dots, x_p(\lambda)\}$ [illegible] $\{y_1(\lambda), \dots, y_q(\lambda)\}$ [illegible] $P(\lambda)$ [illegible] $\deg(x_i)$ [illegible] $\varepsilon_i$ [illegible] $i = 1, \dots, p$ [illegible] $\deg(y_j)$ [illegible] $\eta_j$ [illegible] $j = 1, \dots, q$

[illegible]. We prove the result for the column minimal indices. For the row minimal indices simply use $P^T(\lambda)$ and invoke the result for column minimal indices. Let us recall [8, Chapter XII, p. 38] the relationship between the column minimal indices of $P(\lambda)$ and the polynomial solutions of $P(\lambda)x(\lambda) = 0$. Among all the polynomial solutions of this system of equations we choose a nonzero solution $z_1(\lambda)$ of least degree. This degree is $\varepsilon_1$. Among all the polynomial solutions that are linearly independent of $z_1(\lambda)$ we take a solution $z_2(\lambda)$ of least degree. This degree is $\varepsilon_2$. We continue this process until we get a [illegible] $\{z_1(\lambda), \dots, z_p(\lambda)\}$, i.e., $p = \dim \mathcal{N}(P)$ linearly independent polynomial solutions of $P(\lambda)x(\lambda) = 0$ of degrees $\varepsilon_1 \leq \cdots \leq \varepsilon_p$. A fundamental series of solutions is not uniquely determined, but the degrees of its vectors are, and, as we prove in the next paragraph, every fundamental series of solutions is an ROMB and vice versa.

Let us assume that there exists some index $j$ such that $\deg(x_j) < \varepsilon_j$. Let $j_0$ be the least of these indices, i.e., $\deg(x_{j_0}) < \varepsilon_{j_0}$ and $\deg(x_k) \geq \varepsilon_k$ for $k = 1, \dots, j_0 - 1$. Obviously $j_0 > 1$. Therefore, $\varepsilon_{j_0-1} \leq \deg(x_{j_0-1}) \leq \deg(x_{j_0}) < \varepsilon_{j_0}$. The definition of the minimal indices implies that the linearly independent vectors $\{x_1(\lambda), \dots, x_{j_0}(\lambda)\}$ are linear combinations of $\{z_1(\lambda), \dots, z_{j_0-1}(\lambda)\}$. This is impossible. Then $\deg(x_j) \geq \varepsilon_j$ for all $j = 1, \dots, p$, and, in fact, $\deg(x_j) = \varepsilon_j$ for all $j$, because $\{x_1(\lambda), \dots, x_p(\lambda)\}$ is an ROMB. $\square$

We will also use the following related lemma.

Lemma 2.5. [illegible] $P(\lambda)$ [illegible] (3) [illegible] $\{x_1(\lambda), \dots, x_p(\lambda)\}$ [illegible] $P(\lambda)$ [illegible] $P(\lambda)$ [illegible] $\varepsilon_i$ [illegible] $\{x_1(\lambda), \dots, x_j(\lambda)\}$ [illegible] polynomial coefficients [illegible] $j$ [illegible] $\deg(x_j) \leq \varepsilon_i$ [illegible] $P(\lambda)$ [illegible] $\{x_1(\lambda), \dots, x_p(\lambda)\}$ [illegible] polynomial coefficients [illegible]

[illegible]. The fact that every right null space vector is a linear combination of the mentioned vectors is a straightforward consequence of the definition of minimal indices. The fact that the coefficients are polynomials follows from [6, Main Theorem, p. 495]. $\square$

We will need to ascertain the linear independence of some sets of vector polynomials of $\mathbb{C}^n(\lambda)$. In some situations, this problem can be solved through a standard linear independence problem in $\mathbb{C}^n$. This is shown by Lemma 2.6.

Lemma 2.6. [illegible] $\{v_1(\lambda), \dots, v_r(\lambda)\}$ [illegible] $\mathbb{C}^n(\lambda)$ [illegible]

[illegible]

$$v_i(\lambda) = v_{i0} + \lambda\, v_{i1} + \cdots + \lambda^{d_i}\, v_{id_i} \quad \text{for } 1 \leq i \leq r,$$

[illegible] $v_{ij} \in \mathbb{C}^n$ [illegible] $i, j$ [illegible] $d_i = \deg(v_i(\lambda))$

1. [illegible] $\{v_{10}, \dots, v_{r0}\}$ [illegible] $\mathbb{C}^n$ [illegible] $\{v_1(\lambda), \dots, v_r(\lambda)\}$ [illegible] $\mathbb{C}^n(\lambda)$
2. [illegible] $\{v_{1d_1}, \dots, v_{rd_r}\}$ [illegible] $\mathbb{C}^n$ [illegible] $\{v_1(\lambda), \dots, v_r(\lambda)\}$ [illegible] $\mathbb{C}^n(\lambda)$

[illegible] To prove the first item, the linear combination

$$\alpha_1(\lambda)\, v_1(\lambda) + \cdots + \alpha_r(\lambda) v_r(\lambda) = 0 \tag{6}$$

is considered, where $\alpha_i(\lambda)$, $1 \leq i \leq r$, can be chosen to be polynomials, because if they were rational functions, one could multiply by their denominators. Let us express these polynomials as

$$\alpha_i(\lambda) = \alpha_{i0} + \lambda\, \alpha_{i1} + \cdots + \lambda^{t_i}\, \alpha_{it_i} \quad \text{for } 1 \leq i \leq r,$$

where $\alpha_{ij} \in \mathbb{C}$ for all $i, j$. Therefore, the coefficient vector of the term of degree zero in (6) is

$$\sum_{i=1}^{r} \alpha_{i0} v_{i0} = 0.$$

If $\{v_{10}, \dots, v_{r0}\}$ is a linearly independent set in $\mathbb{C}^n$, then $\alpha_{10} = \alpha_{20} = \cdots = \alpha_{r0} = 0$. Thus, the coefficient vector of the term of degree one in (6) is $\sum_{i=1}^{r} \alpha_{i1} v_{i0} = 0$; this implies $\alpha_{11} = \alpha_{21} = \cdots = \alpha_{r1} = 0$. A simple inductive argument completes the proof of the first item. To prove the second item one simply begins with the coefficient of the term with greatest degree, and performs downward the inductive step. ☐

We finish this section with another technical result on the linear independence of vector polynomials.

LEMMA 2.7. [illegible] $\{z_1(\lambda), \dots, z_k(\lambda)\}$ $k < n$ [illegible] $\mathbb{C}^n(\lambda)$ [illegible] $\{z_1'(\lambda), \dots, z_l'(\lambda)\}$ [illegible] $\mathbb{C}^n(\lambda)$ [illegible] $k + l \leq n$ [illegible] $\operatorname{rank}[z_1(\lambda)| \dots |z_k(\lambda)|z_1'(\lambda)| \dots |z_l'(\lambda)] = k$ [illegible] $\{u_1, \dots, u_n\}$ [illegible] $\mathbb{C}^n$ [illegible] $(u_i)_j = \delta_{ij}$ [illegible] $l$ [illegible] $u_{j_1}, \dots, u_{j_l}$ [illegible] $\{z_1(\lambda), \dots, z_k(\lambda), z_1'(\lambda) + \alpha_1 u_{j_1}, \dots, z_l'(\lambda) + \alpha_l u_{j_l}\}$ [illegible] $\mathbb{C}^n(\lambda)$ [illegible] $\alpha_1, \dots, \alpha_l$

[illegible] There exists at least one $u_{j_1}$ such that $\{z_1(\lambda), \dots, z_k(\lambda), u_{j_1}\}$ is linearly independent because, otherwise, all the vectors in $\{u_1, \dots, u_n\}$ would be linear combinations of $\{z_1(\lambda), \dots, z_k(\lambda)\}$. This is impossible because $\{u_1, \dots, u_n\}$ is also a basis of $\mathbb{C}^n(\lambda)$ and $k < n$. This argument can be successively applied to prove that there exist $u_{j_1}, \dots, u_{j_l}$ vectors of the canonical basis such that $\{z_1(\lambda), \dots, z_k(\lambda), u_{j_1}, \dots, u_{j_l}\}$ is linearly independent. Thus $\{z_1(\lambda), \dots, z_k(\lambda), \alpha_1 u_{j_1}, \dots, \alpha_l u_{j_l}\}$ is linearly independent for all nonzero complex numbers $\alpha_1, \dots, \alpha_l$. Notice that the assumption $\operatorname{rank}[z_1(\lambda)| \dots |z_k(\lambda)|z_1'(\lambda)| \dots |z_l'(\lambda)] = k$ implies that the vectors $z_i'(\lambda)$ are linear combinations of $\{z_1(\lambda), \dots, z_k(\lambda)\}$ with coefficients in $\mathbb{C}(\lambda)$. Therefore, elementary column operations can be used to transform the matrix $[z_1(\lambda)| \dots |z_k(\lambda)|\alpha_1 u_{j_1}| \dots |\alpha_l u_{j_l}]$ into $[z_1(\lambda)| \dots |z_k(\lambda)|z_1'(\lambda) + \alpha_1 u_{j_1}| \dots |z_l'(\lambda) + \alpha_l u_{j_l}]$. This does not change the rank of the matrix, which proves the result. ☐

**2.3. Expansion of a pencil as sum of rank-one pencils.** The expansion presented in Lemma 2.8 will play a key role in this paper.

LEMMA 2.8. [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\rho$ [illegible] $\widetilde{\varepsilon}$ [illegible] $Q(\lambda)$ [illegible]

$$Q(\lambda) = v_1(\lambda)w_1(\lambda)^T + \cdots + v_\rho(\lambda)w_\rho(\lambda)^T, \tag{7}$$

[illegible]

(i) $\{v_1(\lambda), \dots, v_\rho(\lambda)\}$ [illegible] $\mathbb{C}^m(\lambda)$ [illegible]

(ii) $\{w_1(\lambda), \dots, w_\rho(\lambda)\}$ [illegible] $\mathbb{C}^n(\lambda)$ [illegible]

(iii) [illegible] $v_i(\lambda)w_i(\lambda)^T$ $1 \le i \le \rho$ [illegible] $m \times n$ [illegible] $v_i(\lambda)$ [illegible] $w_i(\lambda)$ [illegible]

(iv) [illegible] $\widetilde{\varepsilon}$ [illegible] $w_1(\lambda), \dots, w_\rho(\lambda)$ [illegible] $w_i(\lambda)$ [illegible]

[illegible] (7) [illegible] right decomposition of $Q(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $\rho$ [illegible] $\widetilde{\varepsilon}$ [illegible] $w_1(\lambda), \dots, w_\rho(\lambda)$ [illegible]

[illegible]. The result is a direct consequence of the KCF. Let the KCF of $Q$ be

$$\mathcal{K}_Q(\lambda) = \operatorname{diag}(L_{\widetilde{\varepsilon}_1}, \dots, L_{\widetilde{\varepsilon}_h}, L_{\widetilde{\eta}_1}^T, \dots, L_{\widetilde{\eta}_l}^T, \mathcal{J}_Q),$$

where $\mathcal{J}_Q$ is the regular structure of $Q(\lambda)$ and there exist two nonsingular matrices, $X$ and $Y$, such that $Q(\lambda) = X\mathcal{K}_Q(\lambda)Y$. Now, notice that a block $L_{\widetilde{\varepsilon}_i}$ can be expanded as a sum of $\widetilde{\varepsilon}_i$ rank-one pencils,

$$\begin{bmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} \lambda & 1 & \dots & 0 \end{bmatrix} + \cdots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 0 & \dots & \lambda & 1 \end{bmatrix},$$

where the row (column) vectors have degree equal to one (zero). An expansion for a block $L_{\widetilde{\eta}_j}^T$ is obtained by transposition, but now the column (row) vectors have degree equal to one (zero). For the Jordan blocks in $\mathcal{J}_Q$, corresponding to finite or infinite eigenvalues, similar expansions with row vectors of degree zero are possible. All these expansions can be combined with $Q(\lambda) = X\mathcal{K}_Q(\lambda)Y$ to prove straightforwardly the four items of the lemma.

Let us prove now the fact that any other decomposition of $Q(\lambda)$ as a sum of $\rho$ rank-one matrix pencils contains at least $\widetilde{\varepsilon}$ vectors among $w_1(\lambda), \dots, w_\rho(\lambda)$ with degree exactly one. Notice that the set of solutions of $Q(\lambda)x(\lambda) = 0$ is equal to the set of solutions of $[w_1(\lambda), \dots, w_\rho(\lambda)]^T x(\lambda) = 0$, and therefore the column minimal indices of the pencils $Q(\lambda)$ and $D_0 + \lambda D_1 \equiv [w_1(\lambda), \dots, w_\rho(\lambda)]^T$ are equal. If there were less than $\widetilde{\varepsilon}$ vectors among $w_1(\lambda), \dots, w_\rho(\lambda)$ with degree exactly one, then $\operatorname{rank}(D_1) < \widetilde{\varepsilon}$. This implies that the matrix coefficient of $\lambda$ in the KCF of $[w_1(\lambda), \dots, w_\rho(\lambda)]^T$ has also rank smaller than $\widetilde{\varepsilon}$. This is in contradiction with $\widetilde{\varepsilon}$ being the sum of its column minimal indices. □

[illegible] 1. A result similar to that in Lemma 2.8 can be obtained by considering the sum of the row (or left) minimal indices, $\widetilde{\eta}$, of $Q(\lambda)$ and choosing the column vectors of the expansions of the Jordan blocks in $\mathcal{J}_Q$ to be of degree zero. In this case, we will consider a [illegible] of $Q(\lambda)$:

$$Q(\lambda) = \widehat{v}_1(\lambda)\widehat{w}_1(\lambda)^T + \cdots + \widehat{v}_\rho(\lambda)\widehat{w}_\rho(\lambda)^T, \tag{8}$$

where the vectors $\{\widehat{v}_1(\lambda), \ldots, \widehat{v}_\rho(\lambda)\}$ and $\{\widehat{w}_1(\lambda), \ldots, \widehat{w}_\rho(\lambda)\}$ have the properties appearing in items (i), (ii), and (iii) of Lemma 2.8, but (iv) is replaced by "there are $\widetilde{\eta}$ vectors among $\{\widehat{v}_i(\lambda), \ldots, \widehat{v}_\rho(\lambda)\}$ with degree exactly one, and the remaining vectors are of degree zero." Notice that left and right decompositions are not unique. Besides, a left decomposition of $Q(\lambda)$ may not be simultaneously a right decomposition.

[illegible] 1. Let us show right and left decompositions of a pencil with $\widetilde{\varepsilon} = 1$, $\widetilde{\eta} = 0$, and $\rho = 2$:

$$\begin{bmatrix} \lambda & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \lambda & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \lambda \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} \lambda & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \lambda & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & \lambda \end{bmatrix}.$$

### 2.4. Jordan blocks, invariant polynomials, elementary divisors, and dual pencils.

Given an arbitrary $m \times n$ complex matrix pencil $P(\lambda)$ with rank $r$, there exist two matrix polynomials $U(\lambda)$ and $V(\lambda)$ with dimensions $m \times m$ and $n \times n$, respectively, and nonzero constant determinants, such that

$$U(\lambda)P(\lambda)V(\lambda) = \operatorname{diag}(h_1(P), \ldots, h_r(P), 0, \ldots, 0), \tag{9}$$

where $h_i(P)$ are nonzero monic polynomials in $\lambda$ satisfying $h_i(P) | h_{i+1}(P)$; i.e., $h_i(P)$ divides $h_{i+1}(P)$ for $i = 1, \ldots, r-1$ [8, Chapter VI]. These polynomials are called the [illegible] of $P(\lambda)$, and the diagonal matrix in the right-hand side of (9) is called the [illegible] of $P(\lambda)$. This form is unique and, in fact, [illegible] and not only for pencils. If each

$$h_i(P) = (\lambda - \lambda_1)^{\nu_{i1}} \cdot \cdots \cdot (\lambda - \lambda_d)^{\nu_{id}} \quad \text{for } i = 1, \ldots, r \tag{10}$$

is decomposed in powers of different irreducible factors, then those factors among $(\lambda-\lambda_1)^{\nu_{11}}, \ldots, (\lambda-\lambda_d)^{\nu_{1d}}, \ldots, (\lambda-\lambda_1)^{\nu_{r1}}, \ldots, (\lambda-\lambda_d)^{\nu_{rd}}$ with $\nu_{ij} > 0$ are called the [illegible] $P(\lambda)$. There exists a close relationship between the elementary divisors and the dimensions of the Jordan blocks associated with the finite eigenvalues in the regular structure of the KCF of the pencil $P(\lambda)$. This is revealed by the following result. It is a simple consequence of the theory developed in [8, Chapter VI].

LEMMA 2.9. [illegible] $P(\lambda)$ [illegible] $m \times n$ [illegible] $(\lambda-\lambda_j)^{\nu_{ij}}$ [illegible] $P(\lambda)$ [illegible] $\nu_{ij}$ [illegible] $\lambda_j$ [illegible] $P(\lambda)$ [illegible] $\nu_{ij}$ [illegible] $\lambda_j$ [illegible] $P(\lambda)$ [illegible] $(\lambda - \lambda_j)^{\nu_{ij}}$

The reader should notice that Lemma 2.9 gives no information for the infinite eigenvalue of the pencil $P(\lambda)$. This information can be obtained from the zero eigenvalue of the [illegible] through Lemma 2.10, whose trivial proof is omitted.

LEMMA 2.10. [illegible] $A$ [illegible] $B$ [illegible] $m \times n$ [illegible] $A + \lambda B$ [illegible] $B + \lambda A$ [illegible] $A + \lambda B$ [illegible] $B + \lambda A$ [illegible]

Given an eigenvalue $\lambda_j$ of the pencil $P(\lambda)$, the exponents $0 \leq \nu_{1j} \leq \nu_{2j} \leq \cdots \leq \nu_{rj}$ in (10) are called the [illegible] of $\lambda_j$ relative to $P$, and if a number $\mu$ [illegible] an eigenvalue of $P(\lambda)$, then all its partial multiplicities relative to $P$ are defined as zero [9, p. 331]. Anyway, for any number $\lambda_0$ its partial multiplicities relative to $P$ coincide with the dimensions of the Jordan blocks associated with $\lambda_0$ in the regular structure of the KCF of $P(\lambda)$, whenever Jordan blocks of zero dimension are admitted as nonexisting blocks. This also holds for the infinite eigenvalue. The partial multiplicities of an eigenvalue $\lambda_0$, [illegible], of $P(\lambda)$ with $g$ associated Jordan blocks in the KCF are usually arranged in an infinite sequence called [illegible] of $\lambda_0$ relative to $P(\lambda)$. This sequence is

$$\mathcal{S}_P(\lambda_0) = (n_g(\lambda_0), n_{g-1}(\lambda_0), \ldots, n_1(\lambda_0), 0, \ldots),$$

where $n_g(\lambda_0) \geq n_{g-1}(\lambda_0) \geq \cdots \geq n_1(\lambda_0)$ are the dimensions of the Jordan blocks associated with $\lambda_0$ in the KCF of $P(\lambda)$. Notice that in the case when $\lambda_0$ [illegible] $P(\lambda)$, all the terms in $\mathcal{S}_P(\lambda_0)$ are equal to zero.

The concepts of partial multiplicities and Segre characteristics are also valid for [illegible]. The eigenvalues of a matrix polynomial can be defined as the roots of its invariant polynomials. Given two matrix polynomials $P(\lambda)$ and $Q(\lambda)$, we write

$$\mathcal{S}_P(\lambda_0) \geq \mathcal{S}_Q(\lambda_0) \quad \text{if} \quad (\mathcal{S}_P(\lambda_0))_i \geq (\mathcal{S}_Q(\lambda_0))_i \quad \text{for all } i > 0;$$

i.e., the inequality holds for each entry in the Segre characteristics.

The Smith canonical form (9) allows us to express every matrix polynomial $P(\lambda)$ of rank $r$ as

$$P(\lambda) = h_1(P)\, a_1(\lambda) z_1^T(\lambda) + \cdots + h_r(P)\, a_r(\lambda) z_r^T(\lambda), \tag{11}$$

where $h_1(P), \ldots, h_r(P)$ are its invariant polynomials and $a_i(\lambda)$ and $z_i(\lambda)$ are vector polynomials. Besides, according to (9), the vectors $a_i(\lambda)$ and $z_i(\lambda)$ are, respectively, the columns of $U^{-1}(\lambda)$ and $V^{-T}(\lambda)$. This implies that neither $a_i(\lambda)$ nor $z_i(\lambda)$ can be written as the product of a scalar polynomial of degree greater than zero times a vector polynomial, because the matrices $U^{-1}(\lambda)$ and $V^{-T}(\lambda)$ are matrix polynomials with constant nonzero determinants. Notice that in the case when $P(\lambda)$ [illegible] (11) [illegible] (7), in general, because the summands $h_i(P)\, a_i(\lambda) z_i^T(\lambda)$ have, in general, degree larger than one.

The KCF of the direct sum, $P(\lambda) \oplus Q(\lambda)$, of two pencils $P(\lambda)$ and $Q(\lambda)$ is the direct sum of the KCFs of $P(\lambda)$ and $Q(\lambda)$, up to some permutations of the diagonal blocks. Therefore, the Segre characteristic of $\lambda_0$ relative to $P(\lambda) \oplus Q(\lambda)$ is obtained simply by putting together the Segre characteristics of $\lambda_0$ relative to $P(\lambda)$ and to $Q(\lambda)$, and then reordering the resulting sequence. The same holds in the case when $P(\lambda)$and $Q(\lambda)$ are general matrix polynomials. This follows from [8, Chapter VI, Theorem 5].

**3. Low rank assumptions: Meaning and genericity.** Throughout this work we will deal with three $m \times n$ complex pencils: the fixed [illegible] pencil $P(\lambda)$, the [illegible] pencil $Q(\lambda)$, and the [illegible] pencil $(P+Q)(\lambda)$. The pencil $P(\lambda)$ does not have full rank, and its KCF will always be assumed to be known; it will be denoted by (3). We will frequently omit the variable $\lambda$ when there is no risk of confusion.

As announced in the Introduction, the set of perturbations we considered is the set of pencils

$$\mathcal{C} = \{Q(\lambda) \, : \, \operatorname{rank}(Q) = \rho\}, \tag{12}$$

where $\rho > 0$ is an integer such that

$$\operatorname{rank}(P) + \rho \leq \min\{m, n\} \tag{13}$$

and $\rho \leq \operatorname{rank}(P)$. These are the two low rank conditions imposed on the set of perturbations. Notice also that $\rho > 0$, and (13) implies that $\operatorname{rank}(P) < \min\{m, n\}$, i.e., that $P(\lambda)$ does not have full rank.

A key result in this work is that the property

$$\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) \tag{14}$$

is generic in the set $\mathcal{C}$. This is rigorously proved in Theorem 3.1 below. By combining this result with the identity (4), one can say that, for perturbations in the set $\mathcal{C}$, the perturbed pencils $(P+Q)(\lambda)$ have generically $n - \operatorname{rank}(P) - \rho$ column minimal indices and $m - \operatorname{rank}(P) - \rho$ row minimal indices. See Corollary 3.2 below on this point.

Notice that, taking into account that $P+Q$ is an $m \times n$ pencil, the condition (14) implies $\operatorname{rank}(P) + \operatorname{rank}(Q) \leq \min\{m, n\}$, i.e., the assumption (13), and that $P(\lambda)$ does not have full rank for $Q(\lambda) \neq 0$, i.e., for nontrivial perturbations. These facts and the genericity of (14) in $\mathcal{C}$ lead us to impose $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$ in most of the lemmas and theorems we prove, without explicitly mentioning the initial low rank condition (13).

Section 5 is devoted to studying the generic column minimal indices of $(P+Q)(\lambda)$. In section 5

$$\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n \tag{15}$$

is assumed as a hypothesis in most of the results. Notice that (15) is implied by (14) only if $\min\{m, n\} = m < n$; otherwise (15) is an additional assumption. The reason for assuming (15) in section 5 is that, according to (4), the number of column minimal indices of $P+Q$ is $n - \operatorname{rank}(P+Q)$, which is zero if $\operatorname{rank}(P+Q) = n$. Therefore the study of the generic column minimal indices of $P+Q$ makes sense only if (15) holds. In the case of the row minimal indices, $m$ instead of $n$ has to be used in (15).

Finally, let us comment on the additional low rank assumption,

$$\rho = \operatorname{rank}(Q) \leq \operatorname{rank}(P).$$

This assumption is very natural for considering $Q(\lambda)$ as a low rank perturbation of $P(\lambda)$, and it is essential to guarantee that other hypotheses used in the study of the minimal indices of $P+Q$ are really [illegible]. This will be discussed in subsection 5.5.

### 3.1. Genericity of the assumption $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$. Number of minimal indices of $P+Q$.

The purpose of this section is to present a rigorous proof of the genericity of the most pervasive and crucial assumption in this work. This assumption determines the generic number of row and column minimal indices of the perturbed pencil $(P+Q)(\lambda)$.

THEOREM 3.1. [illegible] $P(\lambda)$ [illegible] $m \times n$ [illegible] $\rho$ [illegible] $\operatorname{rank}(P) + \rho \leq \min\{m,n\}$ [illegible] $m \times n$ [illegible]

$$\mathcal{G} = \{Q(\lambda)\ m \times n \text{ [illegible]} : \operatorname{rank}(Q) = \rho \text{ [illegible]} \quad \operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)\}$$

[illegible] $m \times n$ [illegible] $\rho$

[illegible]. First, let us prove that $\mathcal{G}$ is dense in the set of pencils with rank $\rho$. Notice that $\operatorname{rank}(P+E) \leq \operatorname{rank}(P) + \operatorname{rank}(E)$ for every pencil $E(\lambda)$. Therefore, we have to prove that for every pencil $E(\lambda)$ with rank $\rho$ and $\operatorname{rank}(P+E) < \operatorname{rank}(P) + \operatorname{rank}(E)$ there exists a sequence $\{Q^{(t)}(\lambda)\}_{t=1}^{\infty} \subset \mathcal{G}$ whose limit is $E(\lambda)$. Let $r \equiv \operatorname{rank}(P)$. According to (7), we can write

$$\begin{aligned} P(\lambda) &= v_1(\lambda)w_1(\lambda)^T + \cdots + v_r(\lambda)w_r(\lambda)^T, \\ E(\lambda) &= a_1(\lambda)b_1(\lambda)^T + \cdots + a_\rho(\lambda)b_\rho(\lambda)^T, \end{aligned}$$

and

$$(P+E)(\lambda) = [v_1|\dots|v_r|a_1|\dots|a_\rho]\ [w_1|\dots|w_r|b_1|\dots|b_\rho]^T,$$

where we have omitted some $\lambda$'s for simplicity. Elementary arguments show that $\operatorname{rank}(P+E) < \operatorname{rank}(P) + \operatorname{rank}(E) = r + \rho$ if and only if $\operatorname{rank}[v_1|\dots|v_r|a_1|\dots|a_\rho] < r+\rho$ or $\operatorname{rank}[w_1|\dots|w_r|b_1|\dots|b_\rho] < r+\rho$. Suppose that $\operatorname{rank}[v_1|\dots|v_r|a_1|\dots|a_\rho] < r+\rho$. This implies, due to the fact that the set $\{v_1,\dots,v_r\}$ is linearly independent, that

(i) $\operatorname{rank}[v_1|\dots|v_r|a_1|\dots|a_\rho] = r + \widehat{\rho}$ with $0 \leq \widehat{\rho} < \rho$; and

(ii) the vectors $a_1,\dots,a_\rho$ can be reordered as $a_{i_1},\dots,a_{i_{\widehat{\rho}}}, a_{k_1},\dots,a_{k_{\rho-\widehat{\rho}}}$, where $\{v_1,\dots,v_r,a_{i_1},\dots,a_{i_{\widehat{\rho}}}\}$ is a linearly independent set.

Now, Lemma 2.7 is used to show that there exist $\rho - \widehat{\rho}$ vectors, $u_{j_1},\dots,u_{j_{\rho-\widehat{\rho}}}$, of the canonical bases of $\mathbb{C}^m$ such that, for every $t = 1,2,\dots$,

$$\operatorname{rank}\left[v_1|\dots|v_r|a_{i_1}|\dots|a_{i_{\widehat{\rho}}}|a_{k_1} + \frac{1}{t}u_{j_1}|\dots|a_{k_{\rho-\widehat{\rho}}} + \frac{1}{t}u_{j_{\rho-\widehat{\rho}}}\right] = r + \rho.$$

Let $\{a_1^{(t)},\dots,a_\rho^{(t)}\}$ be the set of vectors that is obtained from $\{a_1,\dots,a_\rho\}$ by replacing $a_{k_1},\dots,a_{k_{\rho-\widehat{\rho}}}$ by $a_{k_1} + \frac{1}{t}u_{j_1},\dots,a_{k_{\rho-\widehat{\rho}}} + \frac{1}{t}u_{j_{\rho-\widehat{\rho}}}$. If $\operatorname{rank}[w_1|\dots|w_r|b_1|\dots|b_\rho] < r+\rho$, we proceed in a similar way to produce a set of vectors $\{b_1^{(t)},\dots,b_\rho^{(t)}\}$. Finally, let us define the sequence of pencils

$$Q^{(t)}(\lambda) = a_1^{(t)}(\lambda)(b_1^{(t)}(\lambda))^T + \cdots + a_\rho^{(t)}(\lambda)(b_\rho^{(t)}(\lambda))^T, \quad t = 1,2,\dots.$$

It is trivial to check that (i) $\lim_{t\to\infty} Q^{(t)}(\lambda) = E(\lambda)$; (ii) $\operatorname{rank}(Q^{(t)}) = \rho$ for all $t$; and (iii) $\operatorname{rank}(P+Q^{(t)}) = \operatorname{rank}(P) + \operatorname{rank}(Q^{(t)})$ for all $t$. This proves that $\mathcal{G}$ is dense.

Now, we will prove that $\mathcal{G}$ is open in the set of matrix pencils with rank $\rho$. To this purpose, let us proceed as follows: as explained in the Introduction, the set of $m \times n$ complex matrix pencils is identified with $\mathbb{C}^{2mn}$, and the set of matrix pencils with rank $\rho$ is a subset $\mathcal{C}$ of $\mathbb{C}^{2mn}$. Thus $\mathcal{G} \subset \mathcal{C} \subset \mathbb{C}^{2mn}$. We consider in $\mathcal{C}$ the [illegible] induced by the usual topology of $\mathbb{C}^{2mn}$, as we explained in the Introduction. Therefore, for proving that $\mathcal{G}$ is open in $\mathcal{C}$, it is sufficient to prove that every $Q(\lambda) \in \mathcal{G}$ is included in an open subset $\mathcal{X}_Q$ of $\mathbb{C}^{2mn}$ such that

$$\operatorname{rank}(P+E) \geq \operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) \quad \text{for all } E \in \mathcal{X}_Q.$$

The reason is that in this case the following hold:

1. $\mathcal{X}_Q \cap \mathcal{C}$ is open in $\mathcal{C}$; and
2. the fact that $\operatorname{rank}(P) + \operatorname{rank}(Q) \leq \operatorname{rank}(P+E) \leq \operatorname{rank}(P) + \operatorname{rank}(E)$ for all $E \in \mathcal{X}_Q$ implies that $\operatorname{rank}(P+E) = \operatorname{rank}(P) + \operatorname{rank}(E)$ for all $E \in \mathcal{X}_Q \cap \mathcal{C}$. This means that $\mathcal{X}_Q \cap \mathcal{C} \subset \mathcal{G}$ and that $Q$ is an interior point of $\mathcal{G}$.

Let us see how $\mathcal{X}_Q \subset \mathbb{C}^{2mn}$ is constructed. Given $Q \in \mathcal{G}$, the equation $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \rho \equiv r + \rho$ implies that the pencil $(P+Q)(\lambda)$ has a $(r+\rho)\times(r+\rho)$ minor that is a polynomial in $\lambda$ with at least one nonzero coefficient. Let $\det(P+Q)(\alpha,\beta)$ be this minor, where the sets $\alpha \subseteq \{1,\ldots,m\}$ and $\beta \subseteq \{1,\ldots,n\}$ denote, respectively, the rows and columns that define the minor. By identifying every pencil $E(\lambda) = E_0 + \lambda E_1$ with an element of $\mathbb{C}^{2mn}$, the coefficients of $\det(P+E)(\alpha,\beta)$ define a continuous function $f(E)$, $f : \mathbb{C}^{2mn} \longrightarrow \mathbb{C}^{r+\rho+1}$, because these coefficients are polynomials in the entries of the complex matrices $E_0$ and $E_1$. Taking into account that $f(Q) \neq 0$, there exists an open ball, $\mathcal{B}$, in $\mathbb{C}^{r+\rho+1}$ whose center is $f(Q)$ and such that $0 \notin \mathcal{B}$. Then we can take $\mathcal{X}_Q = f^{-1}(\mathcal{B})$, because it is open, $f(E) \neq 0$ for all $E \in \mathcal{X}_Q$, and, therefore, $\operatorname{rank}(P+E) \geq r+\rho$ for all $E \in \mathcal{X}_Q$. ☐

As a consequence of Theorem 3.1 and (4) the generic number of row and column minimal indices of $P+Q$ is determined.

Corollary 3.2. [illegible] $P(\lambda)$ [illegible] $m \times n$ [illegible] $p$ [illegible] $q$ [illegible] $\rho$ [illegible] $\operatorname{rank}(P) + \rho \leq \min\{m,n\}$ [illegible] $Q(\lambda)$ [illegible] $\operatorname{rank}(Q) = \rho$ [illegible] $P+Q$ [illegible] $p-\rho$ [illegible] $q-\rho$ [illegible] $m \times n$ [illegible] $\rho$

## 4. The regular structure of the perturbed pencil.

In this section we get information on the regular structure of the KCF of the perturbed pencil $(P+Q)(\lambda)$ in terms of the regular structures of $P(\lambda)$ and $Q(\lambda)$, i.e., $\mathcal{J}_P$ and $\mathcal{J}_Q$. With only the hypothesis $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$, we prove that for every eigenvalue of $P$ or $Q$, the regular structure of $P+Q$ has as least as many blocks as $\mathcal{J}_P \oplus \mathcal{J}_Q$, with dimensions larger than or equal to the dimensions of the blocks in $\mathcal{J}_P \oplus \mathcal{J}_Q$. Besides, other blocks may be present. This is presented in Theorem 4.4, and it is our first major contribution. In section 6, we will see that the generic regular structure of $P+Q$ is precisely $\mathcal{J}_P \oplus \mathcal{J}_Q$ if $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < \min\{m,n\}$.

In this section, the auxiliary lemma, Lemma 4.1, will be used. It appears without proof in [12]. The proof is elementary.

Lemma 4.1 (see [12, p. 799]). [illegible] $D = \operatorname{diag}(z_1,\ldots,z_k)$ [illegible] $k \times k$ [illegible]

$$\det(D+G) = \det G + \sum z_{\nu_1} \ldots z_{\nu_j} \cdot \det \breve{G}(\nu_1,\ldots,\nu_j),$$

[illegible] $j \in \{1,\ldots,k\}$ [illegible] $\nu_1,\ldots,\nu_j$ [illegible] $1 \leq \nu_1 < \cdots < \nu_j \leq k$ [illegible] $\breve{G}(\nu_1,\ldots,\nu_j)$ [illegible] $G$ [illegible] $\nu_1,\ldots,\nu_j$ [illegible] $\det \breve{G}(1,\ldots,k) \equiv 1$

Lemma 4.2 extends [18, Theorem 1] under more stringent assumptions.

Lemma 4.2. [illegible] $\mathcal{L}(\lambda)$ [illegible] $m \times n$ [illegible] $r$ [illegible] $e_1 \leq \cdots \leq e_r$ [illegible] $\lambda_0$ [illegible] $\mathcal{L}(\lambda)$ [illegible] $\mathcal{M}(\lambda)$ [illegible] $e$ [illegible] $\lambda_0$ [illegible] $\mathcal{M}(\lambda)$ [illegible] $\operatorname{rank}(\mathcal{L}+\mathcal{M}) = \operatorname{rank}(\mathcal{L}) + \operatorname{rank}(\mathcal{M})$ [illegible] $e_i < e \leq e_{i+1}$ [illegible] $i \in \{0, 1, \ldots, r\}$ [illegible] $e_0 \equiv -1$ [illegible] $e_{r+1} \equiv \infty$ [illegible] $f_1 \leq \cdots \leq f_{r+1}$ [illegible] $\lambda_0$ [illegible] $(\mathcal{L}+\mathcal{M})(\lambda)$ [illegible]

$$f_1 = e_1\,, \ldots, \, f_i = e_i, \quad e \leq f_{i+1}, \quad e_{i+1} \leq f_{i+2}\,, \ldots\,, \, e_r \leq f_{r+1}.$$

[illegible] 2. Notice that in Lemma 4.2 it is possible that $e_1 = \cdots = e_r = 0$ or that $e = 0$; i.e., $\lambda_0$ may not be an eigenvalue of $\mathcal{L}(\lambda)$ or of $\mathcal{M}(\lambda)$.

[illegible] 4.2 Theorem 1 in [18] implies that

$$e_1 \leq f_2\,, \, e_2 \leq f_3\,, \, \ldots\,, \, e_r \leq f_{r+1}.$$

So, we only need to prove that $f_1 = e_1, \ldots, f_i = e_i, e \leq f_{i+1}$. Let $U(\lambda)$ and $V(\lambda)$ be the matrix polynomials, with nonzero constant determinants, that transform $\mathcal{L}$ into its Smith normal form, i.e., $U(\lambda)\mathcal{L}(\lambda)V(\lambda) = \operatorname{diag}((\lambda-\lambda_0)^{e_1}p_1(\lambda), \ldots, (\lambda-\lambda_0)^{e_r}p_r(\lambda), 0, \ldots, 0)$, with the polynomials $p_1(\lambda), \ldots, p_r(\lambda)$ such that $p_j(\lambda_0) \neq 0$, for $j = 1, \ldots, r$. Invariant polynomials and partial multiplicities remain unchanged under multiplication by $U(\lambda)$ and $V(\lambda)$; therefore we can focus on the partial multiplicities of the matrix polynomial:

$$\begin{aligned} U(\lambda)(\mathcal{L}+\mathcal{M})(\lambda)V(\lambda) = {} & \operatorname{diag}((\lambda-\lambda_0)^{e_1}p_1(\lambda), \ldots, (\lambda-\lambda_0)^{e_r}p_r(\lambda), 0, \ldots, 0) \\ & + (\lambda-\lambda_0)^{e}x(\lambda)y(\lambda)^T, \end{aligned} \tag{16}$$

where the second term of the right-hand side is $U(\lambda)\mathcal{M}(\lambda)V(\lambda)$ (see (11)).

In the case $e_0 < e \leq e_1$, i.e., $i = 0$, the exponent of the factor $(\lambda - \lambda_0)$ of the greatest common divisor of all $1 \times 1$ minors in (16) is greater than or equal to $e$; thus $e \leq f_1$ by the definition of invariant polynomials [8, Chapter VI, section 3], and the result is proven. Let us assume from now on that $i \geq 1$. In the rest of the proof, we will prove that if $c_k$, $k = 1, \ldots, r+1$, denotes the exponent of the factor $(\lambda - \lambda_0)$ of the greatest common divisor of all $k \times k$ minors in (16), then

$$c_1 = e_1, \quad c_2 = e_1 + e_2\,, \ldots, \quad c_i = e_1 + \cdots + e_i, \quad c_{i+1} \geq e_1 + \cdots + e_i + e. \tag{17}$$

This and the definition of invariant polynomials imply $f_1 = e_1\,, \ldots, \, f_i = e_i\,, \, e \leq f_{i+1}$.

The lowest power of $(\lambda - \lambda_0)$ in a $1 \times 1$ minor of (16) is easily seen to be $e_1$, so $c_1 = e_1$. For $k \geq 2$, let us notice that all the nonzero $k \times k$ minors of (16) must contain at least $k-1$ of the $(1,1), \ldots, (r,r)$ diagonal entries. Then, a nonzero $k \times k$ minor of (16) must be of one of these two types:

(i)

$$\begin{aligned} \det \big( & \operatorname{diag}(\,(\lambda-\lambda_0)^{e_{i_1}}p_{i_1}(\lambda), \ldots, 0, \ldots, (\lambda-\lambda_0)^{e_{i_{k-1}}}p_{i_{k-1}}(\lambda)\,) \\ & + (\lambda-\lambda_0)^{e}[x(\lambda)y(\lambda)^T]_k \big)\,, \end{aligned}$$

(ii)

$$\det \big(\operatorname{diag}((\lambda-\lambda_0)^{e_{i_1}}p_{i_1}(\lambda), \ldots, (\lambda-\lambda_0)^{e_{i_k}}p_{i_k}(\lambda)) + (\lambda-\lambda_0)^{e}[x(\lambda)y(\lambda)^T]_k\big)\,, \tag{18}$$

where $[x(\lambda)y(\lambda)^T]_k$ is some $k \times k$ submatrix of $x(\lambda)y(\lambda)^T$. If we apply Lemma 4.1 to these minors, we see that

(i) every minor of type (i) may be written as

$$(\lambda-\lambda_0)^{e_1+\cdots+e_{k-1}+e}\,q(\lambda), \tag{19}$$

where $q(\lambda)$ is a polynomial;

(ii) every minor of type (ii) may be written as

$$(\lambda-\lambda_0)^{e_1+\cdots+e_{k-1}+\min\{e,e_k\}}\,t(\lambda), \tag{20}$$

where $t(\lambda)$ is a polynomial.

In the case $k=i+1$, these results directly imply that $c_{i+1}\geq e_1+\cdots+e_i+e$. In the case $k\leq i$, (19) and (20) imply $c_k\geq e_1+\cdots+e_k$. Moreover, the equality follows by taking $i_1=1,i_2=2,\ldots,i_k=k$, in (18) and applying Lemma 4.1. ☐

Next we prove a corollary of Lemma 4.2.

Corollary 4.3. [illegible] $\mathcal{L}(\lambda)$ [illegible] $\mathcal{M}(\lambda)$ [illegible] $m\times n$ [illegible] $\operatorname{rank}(\mathcal{M})=1$ [illegible] $\operatorname{rank}(\mathcal{L}+\mathcal{M})=\operatorname{rank}(\mathcal{L})+\operatorname{rank}(\mathcal{M})$ [illegible] $h(\mathcal{M})$ [illegible] $\mathcal{M}(\lambda)$ [illegible]

$$\mathcal{S}_{\mathcal{L}+\mathcal{M}}(\lambda_0)\geq\mathcal{S}_{\mathcal{L}\oplus h(\mathcal{M})}(\lambda_0)=\mathcal{S}_{\mathcal{L}\oplus\mathcal{M}}(\lambda_0)\quad\text{for any complex number }\ \lambda_0.$$

[illegible] Let us use the notation in Lemma 4.2 for the partial multiplicities of $\lambda_0$. The partial multiplicities of $\lambda_0$ relative to $\mathcal{L}\oplus h(\mathcal{M})$ are $e_1\leq\cdots\leq e_i<e\leq e_{i+1}\leq\cdots\leq e_r$, by Theorem 5 in [8, Chapter VI, p. 142]. These are also the partial multiplicities of $\lambda_0$ relative to $\mathcal{L}\oplus\mathcal{M}$, by the same argument. Lemma 4.2 implies the inequality $\mathcal{S}_{\mathcal{L}+\mathcal{M}}(\lambda_0)\geq\mathcal{S}_{\mathcal{L}\oplus h(\mathcal{M})}(\lambda_0)$. ☐

Now we prove the main theorem in this section.

Theorem 4.4. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m\times n$ [illegible] $\operatorname{rank}(P+Q)=\operatorname{rank}(P)+\operatorname{rank}(Q)$ [illegible] $\lambda_0$ [illegible] $\mathcal{S}_{P+Q}(\lambda_0)\geq\mathcal{S}_{P\oplus Q}(\lambda_0)$ [illegible] $\lambda_0$ [illegible] $P(\lambda)$ [illegible] $\lambda_0$ [illegible] $Q(\lambda)$ [illegible] $\lambda_0$ [illegible] $(P+Q)(\lambda)$

[illegible] We consider only finite numbers $\lambda_0$. The result for the infinite eigenvalue follows from considering the zero eigenvalue in the dual pencils of $P(\lambda)$ and $Q(\lambda)$. According to (11), $Q(\lambda)$ can be expressed as

$$Q(\lambda)=h_1(Q)\,b_1(\lambda)c_1^T(\lambda)+\cdots+h_\rho(Q)\,b_\rho(\lambda)c_\rho^T(\lambda),$$

where $\rho\equiv\operatorname{rank}(Q)$ and where $h_1(Q),\ldots,h_\rho(Q)$ are the invariant polynomials of $Q(\lambda)$. The property $\operatorname{rank}(P+Q)=\operatorname{rank}(P)+\operatorname{rank}(Q)$ implies that

$$\begin{aligned}&\operatorname{rank}(P+h_1(Q)\,b_1c_1^T+\cdots+h_k(Q)\,b_kc_k^T)\\&\quad=\operatorname{rank}(P+h_1(Q)\,b_1c_1^T+\cdots+h_{k-1}(Q)\,b_{k-1}c_{k-1}^T)+\operatorname{rank}(h_k(Q)\,b_kc_k^T),\end{aligned}\tag{21}$$

for $k=1,\ldots,\rho$. We have omitted the variable $\lambda$ for the sake of simplicity. Therefore, Corollary 4.3 can be applied $\rho$ times to prove

$$\begin{aligned}\mathcal{S}_{P+Q}(\lambda_0)&\geq\mathcal{S}_{(P+h_1(Q)b_1c_1^T+\cdots+h_{\rho-1}(Q)b_{\rho-1}c_{\rho-1}^T)\oplus h_\rho(Q)}(\lambda_0)\geq\cdots\\&\geq\mathcal{S}_{P\oplus h_1(Q)\oplus\cdots\oplus h_\rho(Q)}(\lambda_0),\end{aligned}$$

where we have used $(A+B)\oplus C=(A\oplus C)+(B\oplus 0)$. Finally, Theorem 5 in [8, Chapter VI, p. 142] implies $\mathcal{S}_{P\oplus h_1(Q)\oplus\cdots\oplus h_\rho(Q)}(\lambda_0)=\mathcal{S}_{P\oplus Q}(\lambda_0)$. ☐

**5. The minimal indices of the perturbed pencil.** The purpose of this section is to determine the minimal indices of the perturbed pencil $(P+Q)(\lambda)$ in terms of data of $P(\lambda)$ and $Q(\lambda)$. For the sake of brevity, we will develop the results only for the column minimal indices. A set of counterpart results for the row minimal indices can be obtained just by considering the column minimal indices of the transpose pencil $(P+Q)^T(\lambda)$.

The main result in this section is Theorem 5.8, where the whole set of column minimal indices of $P+Q$ is found for most perturbations $Q$ having a given rank and a given sum of its column minimal indices. The genericity of the hypotheses of Theorem 5.8 is discussed in subsection 5.5. Theorem 5.10 presents some generic information on the column minimal indices of $P+Q$ when only the rank of the perturbation is available.

According to Lemma 2.4, determining the column minimal indices of $P+Q$ is equivalent to finding the degrees of an ROMB of $P+Q$. This ROMB will not be explicitly constructed, but the degrees of its vectors will be precisely determined. Lemma 5.1 is the key result for this task. It allows us to delimit the search for this basis.

LEMMA 5.1. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$ [illegible] $x(\lambda)$ [illegible] $P+Q$ [illegible] $x(\lambda)$ [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $\mathcal{N}(P+Q) = \mathcal{N}(P) \cap \mathcal{N}(Q)$

[illegible] Let $\operatorname{col}(P)$ be the column space of $P(\lambda)$ in $\mathbb{C}^m(\lambda)$. Then,

$$\dim(\operatorname{col}(P+Q)) \leq \dim(\operatorname{col}[P\ Q]) = \dim(\operatorname{col}(P)) + \dim(\operatorname{col}(Q)) - \dim(\operatorname{col}(P) \cap \operatorname{col}(Q)).$$

The assumption $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$ implies that $\operatorname{col}(P) \cap \operatorname{col}(Q) = \{0\}$. If $x(\lambda)$ is a right null space vector of $P+Q$, then $P(\lambda)x(\lambda) = -Q(\lambda)x(\lambda)$. Notice that the vector $z(\lambda) \equiv P(\lambda)x(\lambda) = -Q(\lambda)x(\lambda)$ is a vector in $\operatorname{col}(P) \cap \operatorname{col}(Q)$, and thus $z(\lambda) = 0$ and $P(\lambda)x(\lambda) = Q(\lambda)x(\lambda) = 0$. The converse is trivial. □

We have already remarked in section 3 that if $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) = n$, then the pencil $(P+Q)(\lambda)$ does not have column minimal indices. Therefore, in the rest of this section, it will be assumed $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n$. This implies that $\operatorname{rank}(Q) < p$, where $p$ is the number of column minimal indices of $P$.

### 5.1. Connection polynomials and associated mosaic Toeplitz matrices.

From Lemma 5.1, it is possible to obtain a more specific characterization of the right null space vectors of $P+Q$.

LEMMA 5.2. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n$ $\widetilde{\varepsilon}$ [illegible] $Q(\lambda)$ [illegible] $\{x_1(\lambda), \ldots, x_p(\lambda)\}$ [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] (7) [illegible] $\widetilde{\varepsilon}$ [illegible] $\{w_1(\lambda), \ldots, w_\rho(\lambda)\}$ [illegible] $(P+Q)(\lambda)$ [illegible] $\{x_1(\lambda), \ldots, x_p(\lambda)\}$ [illegible]

$$x(\lambda) = \alpha_1(\lambda)x_1(\lambda) + \cdots + \alpha_p(\lambda)x_p(\lambda) \tag{22}$$

[illegible] $(P+Q)(\lambda)$ [illegible] $\alpha_1(\lambda),\dots,\alpha_p(\lambda)$ [illegible] $\mathbb{C}^p(\lambda)$

$$\begin{aligned} a_{11}(\lambda)\alpha_1(\lambda)+\cdots+a_{1p}(\lambda)\alpha_p(\lambda) &= 0 \\ a_{\rho 1}(\lambda)\alpha_1(\lambda)+\cdots+a_{\rho p}(\lambda)\alpha_p(\lambda) &= 0, \end{aligned} \tag{23}$$

[illegible]

$$a_{ij}(\lambda)=w_i(\lambda)^T x_j(\lambda), \quad i=1,\dots,\rho,\ j=1,\dots,p. \tag{24}$$

[illegible]. A vector polynomial, $x(\lambda)$, is a right null space vector of $P+Q$ if and only if it is a right null space vector of $P$ and $Q$, by Lemma 5.1. $P(\lambda)x(\lambda)=0$ is equivalent to the fact that $x(\lambda)$ is a linear combination of $\{x_1(\lambda),\dots,x_p(\lambda)\}$ with polynomial coefficients, by Lemma 2.5. $Q(\lambda)x(\lambda)=0$ is equivalent, taking into account (7), to $w_1(\lambda)^T x(\lambda)=\cdots=w_\rho(\lambda)^T x(\lambda)=0$, and this is the system of equations (23). ☐

The system of equations (23) is of capital importance in this work, because the set of its solutions allows us to obtain the right null space of $P+Q$ through (22), and we are looking for the degrees of an ROMB of $\mathcal{N}(P+Q)$. Thus, the coefficients $a_{ij}(\lambda)$ of the system (23) play an essential role. They are polynomials in $\lambda$ and link the pencils $P$ and $Q$. They are used so often that we introduce the following definition.

DEFINITION 5.3. [illegible] $\{a_{ij}(\lambda) : i=1,\dots,\rho\ ,\ j=1,\dots,p\}$ [illegible] (24) [illegible] complete set of right connection polynomials [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$

Since neither an ROMB of $P$ nor a right decomposition (7) of $Q$ is unique, a complete set of right connection polynomials of $P$ and $Q$ is not necessarily unique.

[illegible] 3. A left decomposition (8) of $Q(\lambda)$ and an LOMB $\{y_1(\lambda),\dots,y_q(\lambda)\}$ of $P(\lambda)$ can be considered to define a [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$. These are the polynomials

$$b_{ij}(\lambda)=\widehat{v}_i(\lambda)^T y_j(\lambda), \quad i=1,\dots,\rho\ ,\ j=1,\dots,q.$$

These polynomials are needed to obtain the row minimal indices of $(P+Q)(\lambda)$.

Let us denote by $\varepsilon_1\leq\cdots\leq\varepsilon_p$ the column minimal indices of the unperturbed pencil $P(\lambda)$, and by $\widetilde{\varepsilon}$ the sum of the column minimal indices of the perturbation pencil $Q(\lambda)$, as in section 2. Then the degrees of the right connection polynomials of $P$ and $Q$ are bounded as follows:

$$\deg(a_{ij}(\lambda))\leq\begin{cases}\varepsilon_j+1, & i=1,\dots,\widetilde{\varepsilon},\\ \varepsilon_j, & i=\widetilde{\varepsilon}+1,\dots,\rho.\end{cases} \tag{25}$$

For most perturbations $Q(\lambda)$ these inequalities are, in fact, equalities, but this will not be assumed in the subsequent developments. Nevertheless, the generic behavior for the minimal indices of the perturbed pencil $P+Q$ holds under certain conditions that limit the number of right connection polynomials with degrees strictly less than the right-hand side of (25). These generic conditions involve some of the mosaic Toeplitz matrices appearing in the following definition.

DEFINITION 5.4. [illegible] $\varepsilon_1\leq\cdots\leq\varepsilon_p$ [illegible] $P(\lambda)$ $\widetilde{\varepsilon}$ [illegible] $Q(\lambda)$ [illegible] $\{a_{ij}(\lambda) : i=1,\dots,\rho\ ,\ j=1,\dots,p\}$ [illegible] $P$ [illegible]

$Q$ [illegible][1]

$$a_{ij}(\lambda) = a_{ij}^0 + \lambda\, a_{ij}^1 + \cdots + \lambda^{\epsilon_{ij}} a_{ij}^{\epsilon_{ij}}, \quad \text{where} \quad \epsilon_{ij} = \begin{cases} \varepsilon_j + 1, & i = 1, \dots, \widetilde{\varepsilon}, \\ \varepsilon_j, & i = \widetilde{\varepsilon} + 1, \dots, \rho. \end{cases}$$

[illegible] $k$ [illegible] $d$ [illegible] $1 \le k \le p$ [illegible] $d \ge \varepsilon_k - 1$ [illegible] $d$ [illegible] $a_{ij}(\lambda)$ [illegible] $A_k(d)$ [illegible] $\rho$ [illegible] $k$ [illegible] $(s,t)$ [illegible] $s = 1, \dots, \rho$, $t = 1, \dots, k$ [illegible]

$$(A_k(d))_{st} = \underbrace{\begin{bmatrix} a_{st}^0 & & \\ \vdots & \ddots & \\ a_{st}^{\epsilon_{st}} & & a_{st}^0 \\ & \ddots & \vdots \\ & & a_{st}^{\epsilon_{st}} \end{bmatrix}}_{d - \varepsilon_t + 1},$$

[illegible] $d - \varepsilon_t + 1$ [illegible]

$$\epsilon_{st} + 1 + d - \varepsilon_t = \begin{cases} d + 2, & s = 1, \dots, \widetilde{\varepsilon}, \\ d + 1, & s = \widetilde{\varepsilon} + 1, \dots, \rho. \end{cases}$$

[illegible] 4. Notice that in the case $d = \varepsilon_k - 1$ the [illegible]th column of blocks of $A_k(d)$ is formed by matrices having a "number of columns equal to zero," i.e., by empty matrices. This also happens for those columns whose index $j$ satisfies $d = \varepsilon_j - 1$. We understand in this case that $A_k(d)$ has less than $k$ columns of blocks. This convention will simplify the notation and the statements of our results.

The importance of the family of mosaic Toeplitz matrices $A_k(d)$ is made clear by the next lemma, Lemma 5.5. This result extends and complements Lemma 5.2, characterizing right null space vectors of $P + Q$ of a given degree through systems of constant linear equations, i.e., systems of equations in $\mathbb{C}^n$ and not in $\mathbb{C}^n(\lambda)$ as (23). Degrees are the fundamental quantities in this section, because our goal is to get the degrees of the vectors of an ROMB of $P + Q$, i.e., the column minimal indices of $P + Q$. According to Lemma 5.2 all the right null space vectors, $x(\lambda)$, of $P + Q$ are of the form (22), and they have $\deg(x) = \max_{1 \le i \le p} \{\varepsilon_i + \deg(\alpha_i) : \alpha_i(\lambda) \neq 0\}$ [6, Main Theorem, p. 495]. This implies that if $j$ is the largest index such that $\alpha_j(\lambda) \neq 0$, then $\deg(x) \ge \varepsilon_j$. To look for smaller degrees, one has to consider necessarily linear combinations $x(\lambda) = \alpha_1(\lambda) x_1(\lambda) + \cdots + \alpha_k(\lambda) x_k(\lambda)$, with $k < j$. See also Lemma 2.5.

LEMMA 5.5. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\operatorname{rank}(P + Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n$ $\{x_1(\lambda), \dots, x_p(\lambda)\}$ [illegible] $P$ [illegible] $\varepsilon_1 \le \cdots \le \varepsilon_p$ [illegible] $P$ [illegible] $\deg(x_i) = \varepsilon_i$ [illegible]

1. [illegible] $x(\lambda)$ [illegible] $(P + Q)(\lambda)$ [illegible] $d$ [illegible] $x(\lambda) = \alpha_1(\lambda) x_1(\lambda) + \cdots + \alpha_k(\lambda) x_k(\lambda)$ [illegible] $k$ [illegible] $1 \le k \le p$ [illegible] $d \ge \varepsilon_k$ [illegible] $\{\alpha_1(\lambda), \dots, \alpha_k(\lambda)\}$

---

[1] The reader should notice that the superscript notation $a_{ij}^k$ *does not mean* $a_{ij}$ to the $k$th power.

2. [illegible]

$$x(\lambda) = \alpha_1(\lambda)x_1(\lambda) + \cdots + \alpha_k(\lambda)x_k(\lambda) \tag{26}$$

[illegible] $(P+Q)(\lambda)$ [illegible] $d \geq \varepsilon_k$ [illegible] $\alpha_1(\lambda), \dots, \alpha_k(\lambda)$ [illegible]

(i) [illegible] $\alpha_i(\lambda)$ [illegible]

$$\alpha_i(\lambda) = \alpha_{i0} + \lambda\,\alpha_{i1} + \cdots + \lambda^{d-\varepsilon_i}\,\alpha_{i,d-\varepsilon_i} \tag{27}$$

[illegible] $i = 1, \dots, k$ [illegible] $\alpha_{j,d-\varepsilon_j} \neq 0$ [illegible] $j$

(ii) [illegible] $A_k(d)$ [illegible] $d$ [illegible] $P$ [illegible] $Q$ [illegible] $\{x_1(\lambda), \dots, x_p(\lambda)\}$ [illegible] $\alpha_{il}$ [illegible] constant [illegible]

$$A_k(d) \left[\begin{array}{c} \alpha_{10} \\ \\ \alpha_{1,d-\varepsilon_1} \\ \hline \\ \\ \hline \alpha_{k0} \\ \\ \alpha_{k,d-\varepsilon_k} \end{array}\right] = 0. \tag{28}$$

[illegible] (28) [illegible] $\alpha_{1,d-\varepsilon_1} = \cdots = \alpha_{k,d-\varepsilon_k} = 0$ [illegible] $P+Q$ [illegible] (26) [illegible] $d$

[illegible] The proof of the first item is a direct consequence of Lemma 5.2 and the fact that in (22) $\deg(x) = \max_{1\leq i\leq p}\{\varepsilon_i + \deg(\alpha_i) : \alpha_i(\lambda) \neq 0\}$ according to [6, Main Theorem, p. 495]. Once the index $k$ is chosen, the uniqueness of $\{\alpha_1(\lambda), \dots, \alpha_k(\lambda)\}$ follows from the linear independence of $\{x_1(\lambda), \dots, x_k(\lambda)\}$.

The second item follows from (22) and (23) by setting $\alpha_{k+1}(\lambda) = \cdots = \alpha_p(\lambda) = 0$. Notice that (27) simply states that there are no indices $i$, $1 \leq i \leq k$, such that $\deg(\alpha_i) > d - \varepsilon_i$, because this would imply $\deg(x) > d$. The condition $\alpha_{j,d-\varepsilon_j} \neq 0$ for at least one index $j$ guarantees that $\deg(x) = d$. On the other hand, the linear system (28) is the system obtained from (23) by expanding products and sums of polynomials and equating the coefficients to zero. With these remarks in mind, the proof is trivial. ☐

### 5.2. Properties of mosaic Toeplitz matrices.

Lemma 5.6 gathers the properties of mosaic Toeplitz matrices that we will use to deduce the generic minimal indices of the pencil $P+Q$.

Lemma 5.6. [illegible] $\mathcal{T} = \{A_k(d) : 1 \leq k \leq p, d \geq \varepsilon_k - 1\}$ [illegible] 5.4 [illegible]

1. [illegible] $A_k(d)$ [illegible] $\rho(d+1) + \widetilde{\varepsilon}$
2. [illegible] $A_k(d)$ [illegible] $k(d+1) - \sum_{j=1}^{k} \varepsilon_j$
3. $A_k(d)$ [illegible]

$$k > \rho \quad \text{and} \quad d > \frac{\sum_{i=1}^{k} \varepsilon_i + \widetilde{\varepsilon}}{k - \rho} - 1.$$

4. $A_k(d)$ $A_k(d)$ $A_{k'}(d')$ $\mathcal{T}$ $k' \leq k$ $d' \leq d$

5. $A_k(d)$ $A_k(d)$ $A_{k'}(d')$ $\mathcal{T}$ $k' \geq k$ $d' \geq d$

6. $A_k(d)$

$$\text{rank}\begin{bmatrix} a_{11}(\lambda) & a_{12}(\lambda) & \dots & a_{1j}(\lambda) \\ & & & \\ a_{\rho 1}(\lambda) & a_{\rho 2}(\lambda) & \dots & a_{\rho j}(\lambda) \end{bmatrix} = \rho \quad \text{for} \quad j \geq k. \tag{29}$$

The first three items are direct consequences of the number of rows and columns of the blocks appearing in Definition 5.4.

Item 4. Notice that $A_{k-1}(d)$ is obtained from $A_k(d)$ just by erasing the last column of blocks. As a consequence, the columns of $A_{k-1}(d)$ are a subset of the columns of $A_k(d)$. Then, $A_{k-1}(d)$ has full column rank if $A_k(d)$ has full column rank, and, by induction, $A_{k'}(d)$ has full column rank whenever $k' \leq k$.

If $d-1 \geq \varepsilon_k - 1$, then $A_k(d-1)$ is an element of $\mathcal{T}$, and it is obtained from $A_k(d)$ by erasing the last column of each block of $A_k(d)$ to get a certain matrix $A'_k(d)$ and, after that, erasing the last row of each block of $A'_k(d)$ to get $A_k(d-1)$. However, notice that $A'_k(d)$ is of full column rank, and that the last rows of the blocks of $A'_k(d)$ are zero rows; then $A_k(d-1)$ also has full column rank.

If $d-1 < \varepsilon_k - 1$, then $A_k(d-1)$ is not in $\mathcal{T}$. Let $k' < k$ be the largest index such that $d-1 \geq \varepsilon_{k'} - 1$. Then $A_{k'}(d-1)$ is an element of $\mathcal{T}$, and $A_{k'}(d)$ has full column rank. The argument in the paragraph above is applied to prove that $A_{k'}(d-1)$ has full column rank.

Finally, the results above can be combined inductively to prove item 4.

Item 5. Let $k' \geq k$. Then the submatrix of $A_{k'}(d)$ that lies in the first $k(d+1) - \sum_{j=1}^{k} \varepsilon_j$ columns is precisely $A_k(d)$. Therefore, if $A_k(d)$ has full row rank, then $A_{k'}(d)$ also has full row rank. To complete the proof, let us prove that $A_k(d+t)$ has full row rank for any integer $t > 0$ whenever $A_k(d)$ has full row rank. It is enough to prove this statement for $t = 1$ and then to apply an inductive argument. Notice that the submatrix of $A_k(d)$ that contains the last row of each of the row blocks of $A_k(d)$ has linearly independent rows. This means that for the matrix

$$B \equiv \begin{bmatrix} a_{11}^{\epsilon_{11}} & \dots & a_{1k}^{\epsilon_{1k}} \\ \vdots & & \vdots \\ a_{\rho 1}^{\epsilon_{\rho 1}} & \dots & a_{\rho k}^{\epsilon_{\rho k}} \end{bmatrix},$$

$\text{rank}(B) = \rho$. Observe that the matrix $B$ is also the $\rho \times k$ submatrix of $A_k(d+1)$ that lies in the last rows and columns of the blocks of $A_k(d+1)$. If the last rows and columns of the blocks of $A_k(d+1)$ are moved down and back by permutations to the last positions, the rank does not change, and the matrix we get has the structure

$$\begin{bmatrix} A_k(d) & * \\ 0 & B \end{bmatrix}.$$

The rank of this matrix is clearly $\text{rank}(A_k(d)) + \rho = \rho(d+2) + \widetilde{\varepsilon}$, i.e., the number of rows of $A_k(d+1)$. Therefore $A_k(d+1)$ has full row rank.

Item 6. It is enough to prove this property for $j = k$. If the rows of $A_k(d)$ are linearly independent, then the submatrix of $A_k(d)$ that contains the first row of each of the row blocks of $A_k(d)$ has linearly independent rows. This means that

$$\operatorname{rank}\begin{bmatrix} a_{11}^0 & \cdots & a_{1k}^0 \\ \vdots & & \vdots \\ a_{\rho 1}^0 & \cdots & a_{\rho k}^0 \end{bmatrix} = \rho.$$

The result follows by applying Lemma 2.6.1 to the rows of the matrix (29) for $j = k$. ☐

According to [8, Chapter XII, p. 38] (see also the proof of Lemma 2.4 in this paper), the smallest column minimal index of $P + Q$ is the least degree among the degrees of nonzero right null space vectors of $P+Q$. Taking into account Lemma 5.5.2, this smallest minimal index corresponds to the smallest $d$ for which a linear system of the family (28) ($1 \leq k \leq p$) has nonzero solutions with $\alpha_{j,d-\varepsilon_j} \neq 0$ for at least one index $j$. Our intuition here is that solutions of this kind do not exist, generically, if $A_k(d)$ has a number of rows larger than or equal to the number of columns, and they do exist, generically, in the opposite case. This intuition is based on the idea that if the coefficients of the connection polynomials are random for random perturbations pencils $Q$, then the columns of $A_k(d)$ should be linearly independent if $A_k(d)$ has more rows than columns or the same number of rows and columns. Based on this intuition, one can tentatively think that the most likely value of the smallest minimal index of $P + Q$ for random perturbations $Q$ is the smallest $d$ such that some of the $A_k(d)$ has more columns than rows. Of course, these naive arguments have to be supported with rigorous assumptions, but they, together with Lemma 5.6.3, make it natural to consider the following sequence of integer numbers:

$$d_k = \left\lfloor \frac{\sum_{i=1}^{k} \varepsilon_i + \widetilde{\varepsilon}}{k-\rho} \right\rfloor \quad \text{for} \quad k = \rho+1, \ldots, p,$$

where $\lfloor x \rfloor$ denotes the floor function of $x$, i.e., the largest integer that is less than or equal to $x$. Notice that $A_k(d_k)$ exists only if $d_k \geq \varepsilon_k - 1$; in this case Lemma 5.6.3 guarantees that $A_k(d_k)$ has more columns than rows. However, it is not difficult to devise examples for which $d_k < \varepsilon_k - 1$ for some $k$. The natural candidate for the smallest column minimal index of $P + Q$ is $\min_{\rho+1 \leq k \leq p} d_k$. To prove that this is the case under certain generic assumptions, and also to find the rest of the column minimal indices, it is necessary to study the properties of the sequence $\{d_k\}$.

LEMMA 5.7. $0 \leq \varepsilon_1 \leq \cdots \leq \varepsilon_p$ $p$ $\rho$ $\widetilde{\varepsilon}$ $0 < \rho < p$ $0 \leq \widetilde{\varepsilon} \leq \rho$

$$d_k = \left\lfloor \frac{\sum_{i=1}^{k} \varepsilon_i + \widetilde{\varepsilon}}{k-\rho} \right\rfloor \qquad k = \rho+1, \ldots, p. \tag{30}$$

1. $d_{\rho+1} \geq \varepsilon_{\rho+1} \geq \cdots \geq \varepsilon_1$
2. $d_k < d_{k-1}$ $d_k \geq \varepsilon_k \geq \cdots \geq \varepsilon_1$
3. $d_k < d_{k+1}$ $d_k < d_{k+1} \leq d_{k+2} \leq \cdots \leq d_p$
4. $d_k < d_{k+1}$ $d_i < \varepsilon_i$ $i \geq (k+1)$
5.

$$d_{\min} \equiv \min_{\rho+1 \leq k \leq p} d_k.$$

[illegible] $j$ [illegible] $d_j = d_{\min}$, [illegible] $d_j = d_{\min}$ [illegible] $j_1 < j_2$ [illegible]

$$d_j = d_{\min} \quad [illegible] \quad j_1 \leq j \leq j_2 \quad [illegible] \quad d_j > d_{\min} \quad [illegible] \quad j < j_1 \quad [illegible] \quad j_2 < j.$$

[illegible] $d_{j_1} \geq \varepsilon_{j_1}$

6. [illegible] $s$ [illegible] $d_s = d_{\min}$, [illegible] $d_s \geq \varepsilon_s$ [illegible]

$$\varepsilon_k > d_k \geq d_s \quad [illegible] \quad k > s.$$

7. [illegible] $s$ [illegible] $A_s(d_{\min})$, [illegible] $A_s(d_{\min}-1)$ [illegible] 5.4 [illegible]
   (i) $A_s(d_{\min})$ [illegible] $s$ [illegible]
   (ii) $A_s(d_{\min}-1)$ [illegible]
   (iii) [illegible] $k > s$ $A_k(d_{\min})$ [illegible] $A_k(d_{\min}) = A_s(d_{\min})$

Before proving this lemma, we would like to point out that the index $s$ appearing in item 6 will play an essential role in determining the generic column minimal indices of $P + Q$.

[illegible] 5.7. The first item is trivial.

Item 2. Let us consider the integer divisions

$$\sum_{i=1}^{k} \varepsilon_i + \widetilde{\varepsilon} = (k-\rho)d_k + r_k, \quad \text{where} \quad 0 \leq r_k < k - \rho, \tag{31}$$

$$\sum_{i=1}^{k-1} \varepsilon_i + \widetilde{\varepsilon} = (k-1-\rho)d_{k-1} + r_{k-1}, \quad \text{where} \quad 0 \leq r_{k-1} < k - 1 - \rho. \tag{32}$$

Let us subtract (32) from (31) to get

$$\varepsilon_k = (k-\rho-1)(d_k - d_{k-1}) + d_k + r_k - r_{k-1} \leq (k-\rho-1)(d_k - d_{k-1}) + d_k + k - \rho - 1.$$

Thus, $\varepsilon_k \leq (k-\rho-1)(d_k - d_{k-1} + 1) + d_k \leq d_k$. The last step is a consequence of $(d_k - d_{k-1} + 1) \leq 0$ and $(k-1) > \rho$.

Item 3. Let us consider the integer division

$$\sum_{i=1}^{k+1} \varepsilon_i + \widetilde{\varepsilon} = (k+1-\rho)d_{k+1} + r_{k+1}, \quad \text{where} \quad 0 \leq r_{k+1} < k + 1 - \rho. \tag{33}$$

Let us subtract (31) from (33) to get

$$\varepsilon_{k+1} = (k-\rho)(d_{k+1} - d_k) + d_{k+1} + r_{k+1} - r_k \geq (k-\rho) + d_{k+1} + r_{k+1} - r_k > d_{k+1},$$

where we have used that $r_{k+1} - r_k > -(k-\rho)$. Therefore, we have proved that

$$d_k < d_{k+1} \quad \text{implies} \quad \varepsilon_{k+1} > d_{k+1}. \tag{34}$$

Let us consider now an index $l$ such that $l \geq (k+2)$, and the integer division

$$\sum_{i=1}^{l} \varepsilon_i + \widetilde{\varepsilon} = (l-\rho)d_l + r_l, \quad \text{where} \quad 0 \leq r_l < l - \rho. \tag{35}$$

Let us subtract (33) from (35) to get

$$\varepsilon_l + \varepsilon_{l-1} + \cdots + \varepsilon_{k+2} = (d_l - d_{k+1})(l - \rho) + d_{k+1}(l - (k+1)) + r_l - r_{k+1},$$

and then,

$$(\varepsilon_l - d_{k+1}) + (\varepsilon_{l-1} - d_{k+1}) + \cdots + (\varepsilon_{k+2} - d_{k+1}) = (d_l - d_{k+1})(l - \rho) + r_l - r_{k+1}.$$

The inequality (34) implies $(d_l - d_{k+1})(l - \rho) + r_l - r_{k+1} > 0$, and therefore $(d_l - d_{k+1} + 1)(l - \rho) > 0$. Thus, we have proven that

$$d_k < d_{k+1} \quad \text{implies} \quad d_l \geq d_{k+1} \quad \text{for all } l \geq (k+2). \tag{36}$$

This result allows us to prove the more general result appearing in item 3. Let us proceed by contradiction. Assume that $d_{k+1} \leq d_{k+2} \leq \cdots \leq d_p$ is false. This means that there exists an index $l \geq (k+2)$ such that $d_{k+1} \leq d_{k+2} \leq \cdots \leq d_{l-1} > d_l$. Let $j$ be the smallest integer such that $(k+1) \leq j \leq (l-1)$ and $d_j = d_{j+1} = \cdots = d_{l-1}$. Notice that this integer is at least $k+1$, because $d_k < d_{l-1}$ by (36). Then $d_{j-1} < d_j$, and (36) can be applied with $k = j-1$ to see that $d_j \leq d_l$; on the other hand, $d_j = d_{l-1} > d_l$. This is absurd.

Item 4. Let us prove the result by induction. In (34), we have already proven the base case of the induction: $d_{k+1} < \varepsilon_{k+1}$. Let us assume that $d_i < \varepsilon_i$ for some $i \geq (k+1)$. On the other hand, $d_i \leq d_{i+1}$ due to the result in item 3. If $d_i < d_{i+1}$, one can apply (34) with $k = i$ to see that $d_{i+1} < \varepsilon_{i+1}$. Otherwise, $d_i = d_{i+1}$ and $d_{i+1} < \varepsilon_i \leq \varepsilon_{i+1}$.

Item 5. The fact that the indices are consecutive is a direct consequence of item 3. The fact that $d_{j_1} \geq \varepsilon_{j_1}$ is a consequence of items 1 and 2.

Item 6. If there is only one index $s$ such that $d_s = d_{\min}$, the result is a simple consequence of items 4 and 5. Otherwise, let $j_1$ and $j_2$ be the two indices appearing in item 5. If $s = j_2 \leq p$, the result follows again from item 4. If $s < j_2$, then, by definition, $d_s = d_{s+1} < \varepsilon_{s+1} \leq \cdots \leq \varepsilon_{j_2}$. Therefore, $d_k < \varepsilon_k$ for $s+1 \leq k \leq j_2$. Also, by definition, $d_{j_2} < d_{j_2+1}$, and item 4 implies $d_k < \varepsilon_k$ for $k \geq (j_2+1)$.

Item 7. The assertions on the number of rows and columns of $A_s(d_{\min})$ and $A_s(d_{\min}-1)$ follow from Lemma 5.6.3. Let us remember that the $j$th column of blocks of $A_s(d_{\min})$ has $d_{\min} - \varepsilon_j + 1$ columns; therefore, $d_{\min} \geq \varepsilon_s$ guarantees that all the blocks of $A_s(d_{\min})$ have at least one column. Notice that $d_{\min} \geq \varepsilon_s$ also implies that $d_{\min} - 1 \geq \varepsilon_s - 1$; thus $A_s(d_{\min}-1)$ is defined, but some (or all) of its blocks may be empty. Finally, for $j > s$ we know that $\varepsilon_j > d_{\min}$, i.e., $\varepsilon_j - 1 \geq d_{\min}$. This means that $A_k(d_{\min})$, with $k > s$, is not defined unless $\varepsilon_j - 1 = d_{\min}$ for $s+1 \leq j \leq k$, but in this case the $j$th blocks $(A_k(d_{\min}))_{ij}$ are empty matrices. □

**5.3. Generic column minimal indices of $P+Q$.** Now we are in position to find out which are the generic column minimal indices of the perturbed pencil $(P+Q)(\lambda)$, assuming that, apart from the rank, the sum of the column minimal indices of the perturbation is known. This is done in Theorem 5.8, our second major contribution.

THEOREM 5.8. $P(\lambda)$ $Q(\lambda)$ $m \times n$ $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n$ $\rho \equiv \operatorname{rank}(Q)$ $\varepsilon_1 \leq \cdots \leq \varepsilon_p$ $P$ $\widetilde{\varepsilon}$ $Q$ $\{d_{\rho+1}, \ldots, d_p\}$ (30) $d_{\min}$ $s$ $d_s = d_{\min}$ $d_s \geq \varepsilon_s$ $A_s(d_{\min}-1)$ $A_s(d_{\min})$ $s$ $d_{\min} - 1$

[illegible] $d_{\min}$ [illegible] $P$ [illegible] $Q$ [illegible]

$$A_s(d_{\min}-1) \quad \text{[illegible]} \tag{37}$$

$$A_s(d_{\min}) \quad \text{[illegible]} \tag{38}$$

[illegible] $(P+Q)(\lambda)$ [illegible] $p-\rho$ [illegible]

$$\underbrace{d_{\min} = \cdots = d_{\min}}_{s-\rho-\gamma_s} < \underbrace{(d_{\min}+1) = \cdots = (d_{\min}+1)}_{\gamma_s} \leq \varepsilon_{s+1} \leq \cdots \leq \varepsilon_p, \tag{39}$$

[illegible] $\gamma_s$ [illegible] $\sum_{i=1}^{s} \varepsilon_i + \widetilde{\varepsilon}$ [illegible] $s-\rho$

[illegible] In the first place, let us notice that the ordering appearing in (39) is a consequence of Lemma 5.7.6. Also notice that the number of column minimal indices of $P+Q$ is $p-\rho$; this is a simple consequence of (4) and $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$. For the rest of the proof, it is convenient to bear in mind Lemma 2.4 applied to $P+Q$, and the way ROMBs of $P+Q$ are constructed (see the first paragraph in the proof of Lemma 2.4).

Let us begin by proving that there are no column minimal indices of $P+Q$ smaller than $d_{\min}$. Lemma 5.5.1 and Lemma 5.7.6 guarantee that every right null space vector of $P+Q$ with degree $d < d_{\min}$ is a linear combination of the type $x(\lambda) = \alpha_1(\lambda)x_1(\lambda) + \cdots + \alpha_k(\lambda)x_k(\lambda)$ for some $k \leq s$. In this situation, the matrix $A_k(d)$ appearing in (28) has full column rank in the case $A_s(d_{\min}-1)$ has full column rank, by Lemma 5.6.4. The system (28) has only the zero solution and $x(\lambda) = 0$. In the case $A_s(d_{\min}-1)$ is the empty matrix $d_{\min} = \varepsilon_k$ whenever $1 \leq k \leq s$, and there are no nonzero linear combinations of $\{x_1(\lambda), \ldots, x_p(\lambda)\}$ of degree smaller than $\varepsilon_1 = d_{\min}$, because otherwise the smallest column minimal index of $P$ would be less than $\varepsilon_1$.

Our next step is to prove that $d_{\min}$ is a column minimal index of $P+Q$. The system (28) with coefficient matrix $A_s(d_{\min})$ necessarily has nonzero solutions because $A_s(d_{\min})$ has more columns than rows by Lemma 5.7.7. Besides, there are no nonzero solutions with $\alpha_{1,d_{\min}-\varepsilon_1} = \cdots = \alpha_{s,d_{\min}-\varepsilon_s} = 0$, because otherwise the solutions of (28) correspond to right null space vectors of $P+Q$ of degree less than $d_{\min}$, and we already know that they do not exist. This proves that $d_{\min}$ is the smallest column minimal index of $P+Q$.

To see that there are precisely $s-\rho-\gamma_s$ column minimal indices of $P+Q$ equal to $d_{\min}$, we need to find $s-\rho-\gamma_s$ linearly independent right null space vectors of $P+Q$ of degree $d_{\min}$, and to prove that there are no more. Again, Lemma 5.5.1 and Lemma 5.7.6 guarantee that every right null space vector of $P+Q$ with degree $d_{\min}$ is a linear combination of the type

$$x(\lambda) = \alpha_1(\lambda)x_1(\lambda) + \cdots + \alpha_s(\lambda)x_s(\lambda). \tag{40}$$

Notice that the set of solutions of (28) with $A_s(d_{\min})$ as coefficient matrix can be described in terms of a number of free parameters equal to the difference between the number of columns and the number of rows of $A_s(d_{\min})$, i.e.,

$$s(d_{\min}+1) - \sum_{i=1}^{s} \varepsilon_i - \rho(d_{\min}+1) - \widetilde{\varepsilon} = s - \rho - \gamma_s,$$

where Lemma 5.6.1 and Lemma 5.6.2 have been used. This means that the system of linear equations (28) with $A_s(d_{\min})$ has $s-\rho-\gamma_s$ linearly independent solutions, and, by Lemma 5.5, that they correspond to $s-\rho-\gamma_s$ right null space vectors of $P+Q$ of the form (40) of degree $d_{\min}$. Let us denote these vectors by

$$\{z_1(\lambda), z_2(\lambda), \dots, z_{\beta_s}(\lambda)\}, \quad \text{with } \beta_s \equiv s-\rho-\gamma_s. \tag{41}$$

It is clear that any other solution of (28) corresponds to right null space vectors of degree $d_{\min}$ that are linear combinations of (41) with constant coefficients; however, we still need to prove that the vectors $\{z_1(\lambda), z_2(\lambda), \dots, z_{\beta_s}(\lambda)\}$ can be chosen to be linearly independent in $\mathbb{C}^n(\lambda)$. To see this, notice that the $\beta_s$ free parameters of (28) with $A_s(d_{\min})$ may be taken among the $\alpha_{1,d_{\min}-\varepsilon_1}, \dots, \alpha_{s,d_{\min}-\varepsilon_s}$ variables, because the columns of $A_s(d_{\min})$ that do not correspond to these variables are linearly independent, as we have already seen in the paragraph proving that $d_{\min}$ is the smallest minimal index of $P+Q$. By setting the $l$th of these $\beta_s$ variables equal to 1 and the rest equal to 0, and repeating this process for $l=1,\dots,\beta_s$, a set $\mathcal{S}$ of $\beta_s$ linearly independent solutions of (28) may be obtained. Let us denote by

$$a_l = [\alpha^l_{1,d_{\min}-\varepsilon_1}, \dots, \alpha^l_{s,d_{\min}-\varepsilon_s}]^T, \quad l=1,\dots,\beta_s, \tag{42}$$

a vector containing the shown entries of the $l$th solution of (28) in $\mathcal{S}$. The vectors $\{a_1,\dots,a_{\beta_s}\}$ are obviously linearly independent. If (27) and (40) are recalled, the coefficients of the highest degree terms of the vectors (41) corresponding to the $\beta_s$ solutions of (28) in $\mathcal{S}$ are

$$z_{l,d_{\min}} = \alpha^l_{1,d_{\min}-\varepsilon_1} x_{1,\varepsilon_1} + \dots + \alpha^l_{s,d_{\min}-\varepsilon_s} x_{s,\varepsilon_s}, \qquad \text{for } l=1,\dots,\beta_s, \tag{43}$$

where $x_{i,\varepsilon_i}$ is the highest degree coefficient of $x_i(\lambda)$. The vectors $\{x_{1,\varepsilon_1},\dots,x_{s,\varepsilon_s}\}$ are linearly independent in $\mathbb{C}^n$, because $x_1(\lambda),\dots,x_s(\lambda)$ are part of an ROMB and [6, Main Theorem, Item 2, p. 495] can be applied. Therefore, $\{z_{1,d_{\min}},\dots,z_{\beta_s,d_{\min}}\}$ is a linearly independent set, because $[z_{1,d_{\min}},\dots,z_{\beta_s,d_{\min}}] = [x_{1,\varepsilon_1},\dots,x_{s,\varepsilon_s}][a_1,\dots,a_{\beta_s}]$ and the two matrices in the right-hand side have full column rank. Finally, Lemma 2.6.2 implies that $\{z_1(\lambda), z_2(\lambda), \dots, z_{\beta_s}(\lambda)\}$ are linearly independent, and that there are precisely $\beta_s \equiv s-\rho-\gamma_s$ column minimal indices of $P+Q$ equal to $d_{\min}$.

In this paragraph, we prove that there are $\gamma_s$ column minimal indices of $P+Q$ equal to $d_{\min}+1$. At present, we have found a set $\mathcal{C}_1 = \{z_1(\lambda), z_2(\lambda), \dots, z_{\beta_s}(\lambda)\}$ of $s-\rho-\gamma_s$ linearly independent right null space vectors of $P+Q$ of the form (40) and degree $d_{\min}$. However, the fact that $A_s(d_{\min})$ has full row rank, Lemma 5.6.6 with $j=s$, and Lemma 5.2 imply that a maximal linearly independent set of right null space vectors of $P+Q$ of the form (40) has $s-\rho$ vectors. We will prove that the remaining $\gamma_s$ vectors can be chosen to be of degree $d_{\min}+1$. Let us consider the system (28) with coefficient matrix $A_s(d_{\min}+1)$. The matrix $A_s(d_{\min}+1)$ has full row rank because $A_s(d_{\min})$ has full row rank, and Lemma 5.6.5 can be applied. This means that $\operatorname{rank}(A_s(d_{\min}+1)) = \operatorname{rank}(A_s(d_{\min}))+\rho$. Remember that $A_s(d_{\min}+1)$ can be obtained from $A_s(d_{\min})$ by adding one row and one column in the last positions of each block. Therefore, among the $s$ columns of $A_s(d_{\min}+1)$ that are in the last positions of the column blocks, $s-\rho$ are linear combinations of the remaining columns of $A_s(d_{\min}+1)$. Thus, the corresponding variables in the system (28) with $A_s(d_{\min}+1)$ can be taken as some of the free parameters to solve this system.[2] This

---

[2] The reader should notice that the difference between the number of columns and rows of $A_s(d_{\min}+1)$ is $2(s-\rho)-\gamma_s$. Therefore, the system (28) with matrix $A_s(d_{\min}+1)$ has $2(s-\rho)-\gamma_s$ linearly independent solutions, while there are only $s-\rho$ linearly independent right null space vectors of the form (40). This means that linearly independent solutions of (28) do not always correspond to linearly independent right null space vectors (26).

implies that $s-\rho$ free parameters to solve (28) with $A_s(d_{\min}+1)$ may be taken among the $\alpha_{1,(d_{\min}+1)-\varepsilon_1}, \dots, \alpha_{s,(d_{\min}+1)-\varepsilon_s}$ variables. If we proceed with these parameters as in the previous paragraph (arguments around (41)–(43)), we can find a set $\mathcal{C}_2$ of $(s-\rho)$ linearly independent right null space vectors of $P+Q$ of degree exactly $d_{\min}+1$, and of the form (40). Therefore, we can join the set $\mathcal{C}_1$ of $s-\rho-\gamma_s$ vectors of degree $d_{\min}$ with some $\gamma_s$ vectors of $\mathcal{C}_2$, to get a maximal linearly independent set of right null space vectors of $P+Q$ of the form (40). This proves that there exist $\gamma_s$ column minimal indices of $P+Q$ equal to $d_{\min}+1$.

Our last task in proving Theorem 5.8 is to show that the remaining column minimal indices of $P+Q$ are $\varepsilon_{s+1} \leq \cdots \leq \varepsilon_p$. The right null space vectors of $P+Q$ that we have already found, corresponding to minimal indices equal to $d_{\min}$ and to $d_{\min}+1$, constitute a maximal linearly independent set of right null space vectors of $P+Q$ of the form (40). This fact implies that any right null space vector $x(\lambda)$ of $P+Q$ corresponding to the next smallest minimal index has to be necessarily of the form (22) with at least one of the coefficients $\alpha_{s+1}(\lambda), \dots, \alpha_p(\lambda)$ different from zero. Otherwise, it would depend linearly on the right null space vectors corresponding to the minimal indices $d_{\min}$ and $d_{\min}+1$. Thus, according to [6, Main Theorem, p. 495], $\deg(x) = \max_{1\leq i\leq p}\{\varepsilon_i + \deg(\alpha_i) : \alpha_i(\lambda) \neq 0\} \geq \varepsilon_{s+1} \geq d_{\min}+1$, where the last inequality is a consequence of Lemma 5.7.6. Then, the least candidate to the next minimal index is $\varepsilon_{s+1}$. To show that, in fact, $\varepsilon_{s+1}$ is the next column minimal index, we will prove that there is a right null space vector of $P+Q$ of the form

$$x(\lambda) = \alpha_1(\lambda)x_1(\lambda) + \cdots + \alpha_{s+1}(\lambda)x_{s+1}(\lambda), \tag{44}$$

with $\alpha_{s+1}(\lambda) \neq 0$ and $\deg(x) = \varepsilon_{s+1}$, i.e., with $\alpha_{s+1}(\lambda)$ a nonzero constant. This is equivalent to proving that the linear system (28) with coefficient matrix $A_{s+1}(\varepsilon_{s+1})$ has solutions with the last entry different from zero. Notice that $A_{s+1}(\varepsilon_{s+1})$ has full row rank because $A_s(d_{\min})$ has full row rank, $\varepsilon_{s+1} > d_{\min}$ by Lemma 5.7.6, and Lemma 5.6.5 can be applied. Besides, the matrices in the last columns of blocks of $A_{s+1}(\varepsilon_{s+1})$ have only one column. Therefore, if the last column of $A_{s+1}(\varepsilon_{s+1})$ is removed, then $A_s(\varepsilon_{s+1})$ is obtained. However,

$$\text{number of rows of } A_{s+1}(\varepsilon_{s+1}) = \text{number of rows of } A_s(\varepsilon_{s+1}), \tag{45}$$

and $A_s(\varepsilon_{s+1})$ has also full row rank by the same argument; then

$$\operatorname{rank}(A_{s+1}(\varepsilon_{s+1})) = \operatorname{rank}(A_s(\varepsilon_{s+1})). \tag{46}$$

This implies that the last column of $A_{s+1}(\varepsilon_{s+1})$ is a linear combination of its remaining columns. As a consequence the last variable in the linear system (28) with coefficient matrix $A_{s+1}(\varepsilon_{s+1})$ may be considered as free parameter, and therefore it may be different from zero. This proves that the $\varepsilon_{s+1}$ is the next minimal index.

Notice that assumption (38) and Lemma 5.6.6 imply that a maximal linearly independent set of right null space vectors of $P+Q$ of the form (44) has $s+1-\rho$ vectors. Therefore, a maximal linearly independent set of this type has already been found in the previous paragraphs. With this remark in mind, the proof that $\varepsilon_{s+2}$ is the next smallest column minimal index follows step-by-step the proof presented in the previous paragraph for $\varepsilon_{s+1}$ with the corresponding changes of indices. The same holds for proving that $\varepsilon_{s+3}, \dots, \varepsilon_p$ are the remaining column minimal indices of $P+Q$. ☐

**5.4. Application of Theorem 5.8 to an example.** Let us show with an example how to apply Theorem 5.8. Let $P(\lambda)$ be the $5 \times 5$ matrix pencil

$$P(\lambda) = \operatorname{diag}(L_0,\, L_1,\, L_1, L_0^T,\, L_0^T,\, L_0^T) = \begin{bmatrix} 0 & \lambda & 1 & 0 & 0 \\ 0 & 0 & 0 & \lambda & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

with $\varepsilon_1 = 0$, $\varepsilon_2 = \varepsilon_3 = 1$. An ROMB of $P$ is given by

$$x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad x_2 = \begin{bmatrix} 0 \\ 1 \\ -\lambda \\ 0 \\ 0 \end{bmatrix}, \qquad x_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ -\lambda \end{bmatrix}.$$

Consider an arbitrary perturbation $Q$ of $P$ with $\rho = 2$ and $\widetilde{\varepsilon} = 1$. This means that a right decomposition of $Q$ (see (7)) is of the form

$$Q(\lambda) = v_1 w_1^T + v_2 w_2^T,$$

where

$$w_1 = \begin{bmatrix} b_1 + \lambda c_1 \\ b_2 + \lambda c_2 \\ b_3 + \lambda c_3 \\ b_4 + \lambda c_4 \\ b_5 + \lambda c_5 \end{bmatrix}, \qquad w_2 = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{bmatrix},$$

and $b_i$, $c_i$, $d_i \in \mathbb{C}$ for $i = 1, \ldots, 5$. In addition, $\deg(v_1) = 0$ and $\deg(v_2) \leq 1$. Notice that in this example $p - \rho = 3 - 2 = 1$, and so the sequence $\{d_{\rho+1}, \ldots, d_p\}$ has only the element $d_3$. This means that in the conditions of Theorem 5.8 [illegible] $P + Q$ [illegible]

$$d_{\min} = d_3 = \left\lfloor \frac{0+1+1+1}{3-2} \right\rfloor = 3.$$

The matrices $A_s(d_{\min} - 1)$ and $A_s(d_{\min})$ are in this case $A_3(2)$ and $A_3(3)$, respectively. The right connection polynomials associated with the previous data are given by

$$\begin{aligned} a_{11}(\lambda) &= w_1(\lambda)^T x_1 = b_1 + \lambda c_1, & a_{21}(\lambda) &= w_2^T x_1 = d_1, \\ a_{12}(\lambda) &= w_1(\lambda)^T x_2 = b_2 + \lambda(c_2 - b_3) - \lambda^2 c_3, & a_{22}(\lambda) &= w_2^T x_2 = d_2 - \lambda d_3, \\ a_{13}(\lambda) &= w_1(\lambda)^T x_3 = b_4 + \lambda(c_4 - b_5) - \lambda^2 c_5, & a_{23}(\lambda) &= w_2^T x_3 = d_4 - \lambda d_5, \end{aligned}$$

and then the mosaic Toeplitz matrix $A_3(3)$ is the $9 \times 10$ matrix

$$A_3(3) = \left[\begin{array}{cccc|ccc|ccc} b_1 & 0 & 0 & 0 & b_2 & 0 & 0 & b_4 & 0 & 0 \\ c_1 & b_1 & 0 & 0 & c_2 - b_3 & b_2 & 0 & c_4 - b_5 & b_4 & 0 \\ 0 & c_1 & b_1 & 0 & -c_3 & c_2 - b_3 & b_2 & -c_5 & c_4 - b_5 & b_4 \\ 0 & 0 & c_1 & b_1 & 0 & -c_3 & c_2 - b_3 & 0 & -c_5 & c_4 - b_5 \\ 0 & 0 & 0 & c_1 & 0 & 0 & -c_3 & 0 & 0 & -c_5 \\ \hline d_1 & 0 & 0 & 0 & d_2 & 0 & 0 & d_4 & 0 & 0 \\ 0 & d_1 & 0 & 0 & -d_3 & d_2 & 0 & -d_5 & d_4 & 0 \\ 0 & 0 & d_1 & 0 & 0 & -d_3 & d_2 & 0 & -d_5 & d_4 \\ 0 & 0 & 0 & d_1 & 0 & 0 & -d_3 & 0 & 0 & -d_5 \end{array}\right].$$

It can be numerically checked using MATLAB that this matrix has full row rank for random values of $b_i, c_i$, and $d_i$. The $7 \times 7$ matrix $A_3(2)$ is constructed in a similar way, and it can be numerically checked that $A_3(2)$ has full column rank.

**5.5. On the genericity of the assumptions of Theorem 5.8.** The relevance of Theorem 5.8 depends on the [illegible] of its hypotheses, i.e., whether they are satisfied in a dense open subset of the considered set of perturbations. The meaning and genericity of the condition $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n$ was discussed in depth in section 3. The other two essential hypotheses in Theorem 5.8 are (37)–(38). We have checked numerically with MATLAB on a sample of more than 50000 mosaic Toeplitz matrices (Definition 5.4), constructed on random polynomials, that these matrices have full rank. We have run experiments with matrices with more rows than columns, and vice versa. Then to see that (37)–(38) are indeed generic assumptions that hold for almost all perturbations, it remains only to justify that the connection polynomials of $P$ and $Q$ are [illegible] for random perturbations $Q$. In this process the natural assumption

$$\operatorname{rank}(Q) \leq \operatorname{rank}(P), \tag{47}$$

noted in section 3, plays a relevant role. Let us remember Definition 5.3. We can assume in the following argument, without loss of generality, that $P$ is given in KCF. Taking into account that the right null space vector of a column singular block $L_\varepsilon$ can be chosen to be $[1, -\lambda, \lambda^2, \ldots, (-\lambda)^\varepsilon]^T$, the vectors $\{x_1(\lambda), \ldots, x_p(\lambda)\}$ of the ROMB of $P$ can be chosen with the following property: if $(x_j(\lambda))_k \neq 0$ for some $j$, then $(x_{j'}(\lambda))_k = 0$ for $j' \neq j$; i.e., the nonzero entries of every vector correspond to zero entries of the remaining vectors. With this in mind, it seems at a first glance that the coefficients of the connection polynomials (24) are random, because $Q$ being a random perturbation, the vectors $\{w_1(\lambda), \ldots, w_\rho(\lambda)\}$ should be also random. But, according to (7), the vectors $w_i(\lambda)$ are of degree at most one. This means that, putting together the zero and first order coefficients of each $w_i(\lambda)$, we get a set $\mathcal{W}$ with $\rho + \widetilde{\varepsilon}$ vectors of $\mathbb{C}^n$. Notice that $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) \leq \min\{m, n\}$ and (47) imply that $\rho + \widetilde{\varepsilon} \leq n$, and then the vectors in $\mathcal{W}$ are linearly independent for almost all $Q$, and the coefficients of the connection polynomials are really random. But, if $\rho + \widetilde{\varepsilon} > n$, then the set $\mathcal{W}$ is linearly dependent, and some linear dependence may appear among the coefficients of the connection polynomials.[3]

**5.6. When the only information available on the perturbation is its rank.** Theorem 5.8 determines the generic whole set of column minimal indices of the perturbed pencil $(P+Q)(\lambda)$. This set depends on $\widetilde{\varepsilon}$, i.e., the sum of the column minimal indices of the perturbation $Q(\lambda)$. The reason for this dependence can be traced back to the expansion (7), because the properties of (7) depend on $\widetilde{\varepsilon}$. This fact is related to a deeper mathematical result: the set of singular matrix pencils of rank $\rho$ has exactly $\rho + 1$ maximal irreducible components, each of them corresponding to a value of $\widetilde{\varepsilon}$, for $\widetilde{\varepsilon} = 0, \ldots, \rho$ [3]. However, one may want to get some partial information if only the rank of the perturbation $Q(\lambda)$ is known. This partial information is presented in Theorem 5.10. To prove this theorem the following lemma is needed.

LEMMA 5.9. [illegible] $0 \leq \varepsilon_1 \leq \cdots \leq \varepsilon_p$ [illegible] $p$ [illegible] $\rho$ [illegible] $\widetilde{\varepsilon}$ [illegible] $0 < \rho < p$ [illegible] $0 \leq \widetilde{\varepsilon} \leq \rho$ [illegible] $\widetilde{\varepsilon}$ [illegible]

---

[3] Notice that the argument of this paragraph holds if the assumption (47) is replaced by $\rho + \widetilde{\varepsilon} \leq n$, which is fulfilled by a wider set of perturbations. However, this condition is not natural and requires knowing $\widetilde{\varepsilon}$ apart from the rank of the perturbation.

[illegible]

$$d_k(\widetilde{\varepsilon}) = \left\lfloor \frac{\sum_{i=1}^{k} \varepsilon_i + \widetilde{\varepsilon}}{k-\rho} \right\rfloor \qquad k = \rho+1,\dots,p. \tag{48}$$

[illegible] $d_{\min}(\widetilde{\varepsilon}) \equiv \min_{\rho+1\le k\le p} d_k(\widetilde{\varepsilon})$ [illegible] $s(\widetilde{\varepsilon})$ [illegible] $d_{s(\widetilde{\varepsilon})}(\widetilde{\varepsilon}) = d_{\min}(\widetilde{\varepsilon})$ [illegible] $d_{s(\widetilde{\varepsilon})}(\widetilde{\varepsilon}) \ge \varepsilon_{s(\widetilde{\varepsilon})}$ [illegible]

$$d_{\min}(\rho) \ge d_{\min}(\rho-1) \ge \cdots \ge d_{\min}(0) \quad \text{and} \quad s(\rho) \ge s(\rho-1) \ge \cdots \ge s(0).$$

[illegible] Let us prove that $d_{\min}(\widetilde{\varepsilon}') \ge d_{\min}(\widetilde{\varepsilon})$ and $s(\widetilde{\varepsilon}') \ge s(\widetilde{\varepsilon})$, whenever $\widetilde{\varepsilon}' > \widetilde{\varepsilon}$. Notice that $d_k(\widetilde{\varepsilon}') \ge d_k(\widetilde{\varepsilon})$ for $k = \rho+1,\dots,p$. Therefore $d_{\min}(\widetilde{\varepsilon}') \ge d_{\min}(\widetilde{\varepsilon})$. According to Lemma 5.7.6, $\varepsilon_{s(\widetilde{\varepsilon}')+1} > d_{s(\widetilde{\varepsilon}')}(\widetilde{\varepsilon}') \ge d_{s(\widetilde{\varepsilon})}(\widetilde{\varepsilon}) \ge \varepsilon_{s(\widetilde{\varepsilon})}$. This means that $s(\widetilde{\varepsilon}') + 1 > s(\widetilde{\varepsilon})$. ☐

By combining Theorem 5.8 and Lemma 5.9, we can state the generic theorem, Theorem 5.10. We name Theorem 5.10 as generic, because the precise assumptions needed in the theorem are (37)–(38), and they depend on the sum of the column minimal indices of the perturbation $Q$, information that is not available. The only requirement for proving Theorem 5.10 is to notice that if $\rho$ is the rank of $Q$ and $\widetilde{\varepsilon}$ is the sum of the column minimal indices of $Q$, then $0 \le \widetilde{\varepsilon} \le \rho$.

THEOREM 5.10. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < n$ [illegible] $\operatorname{rank}(Q) \le \operatorname{rank}(P)$ [illegible] $\rho \equiv \operatorname{rank}(Q)$ [illegible] $\varepsilon_1 \le \cdots \le \varepsilon_p$ [illegible] $P$ [illegible]

$$d_k' = \left\lfloor \frac{\sum_{i=1}^{k} \varepsilon_i + \rho}{k-\rho} \right\rfloor \qquad k = \rho+1,\dots,p. \tag{49}$$

[illegible] $d_{\min}'$ [illegible] $\{d_k'\}$ [illegible] $s'$ [illegible] $d_{s'}' = d_{\min}'$ [illegible] $d_{s'}' \ge \varepsilon_{s'}$ [illegible] $Q(\lambda)$ [illegible] $\rho$ $(P+Q)(\lambda)$ [illegible] $p-\rho$ [illegible]

1. [illegible] $p-s'$ [illegible] $(P+Q)(\lambda)$ [illegible] $\varepsilon_{s'+1} \le \cdots \le \varepsilon_p$.
2. [illegible] $s'-\rho$ [illegible] $(P+Q)(\lambda)$ $\hat{\varepsilon}_1 \le \cdots \le \hat{\varepsilon}_{s'-\rho}$ [illegible] $\varepsilon_{\rho+j} \le \hat{\varepsilon}_j$ [illegible] $j = 1,\dots,s'-\rho$ [illegible] $\hat{\varepsilon}_1 \le d_{min}'$

## 6. The Kronecker canonical form of perturbed pencils without full rank.

The results presented so far remain valid whenever $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) \le \min\{m,n\}$. This assumption includes the limit case $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) = \min\{m,n\}$, i.e., the case of perturbed pencils $(P+Q)(\lambda)$ with full rank. In this full rank case $P+Q$ does not have row minimal indices if $\operatorname{rank}(P+Q) = m$, and $P+Q$ does not have column minimal indices if $\operatorname{rank}(P+Q) = n$, according to (4). If $\operatorname{rank}(P+Q) = m < n$, the generic column minimal indices of $P+Q$ are described by Theorem 5.8, and if $\operatorname{rank}(P+Q) = n < m$, the generic row minimal indices of $P+Q$ are described by Theorem 5.8 applied on $(P+Q)^T$. Theorem 4.4 also holds in the full rank case and gives partial information on the regular part of $P+Q$.

The purpose of this section is to show that complete information on the generic KCF of $P+Q$ can be obtained for perturbed pencils without full rank, i.e.,

$$\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) < \min\{m,n\}.$$

We will gather the information obtained in Theorems 4.4 and 5.8, together with the counterpart version of Theorem 5.8 for row minimal indices, to fully describe the generic KCF of $(P+Q)(\lambda)$, in terms of the sums of the row and column minimal indices and of the regular structure of the perturbation $Q(\lambda)$. This KCF will be presented in Theorem 6.2. In addition, Theorem 6.3 presents some generic partial information on the KCF of $P+Q$ when $\operatorname{rank}(Q)$ is the only information available on the perturbation.

Lemma 6.1 will allow us to avoid certain redundancy in the hypotheses.

LEMMA 6.1. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\operatorname{rank}(P) + \operatorname{rank}(Q) < \min\{m,n\}$ [illegible] $A_s(d_{\min})$ [illegible] $P$ [illegible] $Q$ [illegible] 5.8 [illegible] $B_t(h_{\min})$ [illegible] 5.8 [illegible] $P^T$ [illegible] $Q^T$ [illegible] $A_s(d_{\min})$ [illegible] $B_t(h_{\min})$ [illegible] $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$

[illegible] From elementary linear algebra we know $\operatorname{rank}(P+Q) \leq \operatorname{rank}(P) + \operatorname{rank}(Q)$. Let us consider right decompositions of $P$ and $Q$ of the kind appearing in (7):

$$P(\lambda) = v_1'(\lambda)w_1'(\lambda)^T + \cdots + v_r'(\lambda)w_r'(\lambda)^T,$$
$$Q(\lambda) = v_1(\lambda)w_1(\lambda)^T + \cdots + v_\rho(\lambda)w_\rho(\lambda)^T,$$

where $r \equiv \operatorname{rank}(P)$ and $\rho \equiv \operatorname{rank}(Q)$. Therefore

$$P + Q = [v_1', \dots, v_r', v_1, \dots, v_\rho]\,[w_1', \dots, w_r', w_1, \dots, w_\rho]^T,$$

where the dependence on $\lambda$ has been omitted. This means that $P+Q$ is the product of an $m \times (r+\rho)$ matrix times an $(r+\rho) \times n$ matrix, with $(r+\rho) < \min\{m,n\}$. Therefore $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$ if and only if

$$\operatorname{rank}[v_1', \dots, v_r', v_1, \dots, v_\rho] = r + \rho \quad \text{and} \quad \operatorname{rank}[w_1', \dots, w_r', w_1, \dots, w_\rho] = r + \rho.$$

Let us prove that if $A_s(d_{\min})$ has full row rank, then $\operatorname{rank}[w_1', \dots, w_r', w_1, \dots, w_\rho] = r + \rho$. If $\operatorname{rank}[w_1', \dots, w_r', w_1, \dots, w_\rho] < r + \rho$, there exists an index $i$ such that $w_i(\lambda)$ is a linear combination of $\{w_1'(\lambda), \dots, w_r'(\lambda), w_1(\lambda), \dots, w_{i-1}(\lambda)\}$ in $\mathbb{C}^n(\lambda)$, i.e.,

$$w_i(\lambda) = \beta_1'(\lambda)w_1'(\lambda) + \cdots + \beta_r'(\lambda)w_r'(\lambda) + \beta_1(\lambda)w_1(\lambda) + \cdots + \beta_{i-1}(\lambda)w_{i-1}(\lambda),$$

for some rational functions $\beta_1'(\lambda), \dots, \beta_{i-1}(\lambda)$. Let us recall (24) and $P(\lambda)x_j(\lambda) = 0$, i.e., $w_k'(\lambda)^T x_j(\lambda) = 0$ for all $k$. Then the right connection polynomials of $P$ and $Q$ satisfy

$$a_{ij}(\lambda) = \beta_1(\lambda)a_{1j}(\lambda) + \cdots + \beta_{i-1}(\lambda)a_{i-1,j}(\lambda) \quad \text{for} \quad j = 1, \dots, p,$$

and the matrix $[a_{kl}(\lambda)]_{1 \leq k \leq \rho}^{1 \leq l \leq p}$ does not have full row rank. But the fact that $A_s(d_{\min})$ has full row rank implies that $\operatorname{rank}[a_{kl}(\lambda)]_{1 \leq k \leq \rho}^{1 \leq l \leq p} = \rho$, by Lemma 5.6.6.

An analogous argument shows that if $B_t(h_{\min})$ has full row rank, then $\operatorname{rank}[v_1', \dots, v_r', v_1, \dots, v_\rho] = r + \rho$. $\square$

We do not explicitly impose $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q)$ in Theorem 6.2 because of Lemma 6.1.

THEOREM 6.2. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\operatorname{rank}(P) + \operatorname{rank}(Q) < \min\{m,n\}$ [illegible] $\rho \equiv \operatorname{rank}(Q)$ [illegible] $\varepsilon_1 \leq \cdots \leq \varepsilon_p$ [illegible] $\eta_1 \leq \cdots \leq \eta_q$ [illegible] $P$ [illegible] $\mathcal{J}_P$ [illegible] $P$ [illegible] $\widetilde{\varepsilon}$ [illegible] $\widetilde{\eta}$ [illegible]

[illegible] $Q$ [illegible] $\mathcal{J}_Q$ [illegible] $Q$ [illegible]

$$d_k = \left\lfloor \frac{\sum_{i=1}^{k} \varepsilon_i + \widetilde{\varepsilon}}{k-\rho} \right\rfloor \quad \text{[illegible]} \quad k = \rho+1,\dots,p \quad \text{[illegible]}$$

$$h_l = \left\lfloor \frac{\sum_{i=1}^{l} \eta_i + \widetilde{\eta}}{l-\rho} \right\rfloor \quad \text{[illegible]} \quad l = \rho+1,\dots,q.$$

[illegible] $d_{\min} = \min_{\rho+1\le k\le p}\{d_k\}$ [illegible] $s$ [illegible] $d_s = d_{\min}$ [illegible] $d_s \ge \varepsilon_s$ [illegible] $h_{\min} = \min_{\rho+1\le l\le q}\{h_l\}$ [illegible] $t$ [illegible] $h_t = h_{\min}$ [illegible] $h_t \ge \eta_t$ [illegible] $A_s(d_{\min}-1)$ [illegible] $A_s(d_{\min})$ $B_t(h_{\min}-1)$ [illegible] $B_t(h_{\min})$ [illegible] $s$ [illegible] $t$ [illegible] $d_{\min}-1$ [illegible] $d_{\min}$ $h_{\min}-1$ [illegible] $h_{\min}$ [illegible] $P$ [illegible] $Q$ [illegible]

$A_s(d_{\min}-1)$ [illegible] $B_t(h_{\min}-1)$ [illegible] $A_s(d_{\min})$ [illegible] $B_t(h_{\min})$ [illegible]

[illegible]

1. $(P+Q)(\lambda)$ [illegible] $p-\rho$ [illegible]

$$d_{\min} = \cdots = d_{\min} \tag{50}$$

$$\underbrace{d_{\min} = \cdots = d_{\min}}_{s-\rho-\gamma_s} < \underbrace{(d_{\min}+1) = \cdots = (d_{\min}+1)}_{\gamma_s} \le \varepsilon_{s+1} \le \cdots \le \varepsilon_p, \tag{50}$$

[illegible] $\gamma_s$ [illegible] $\sum_{i=1}^{s} \varepsilon_i + \widetilde{\varepsilon}$ [illegible] $s-\rho$.

2. $(P+Q)(\lambda)$ [illegible] $q-\rho$ [illegible]

$$\underbrace{h_{\min} = \cdots = h_{\min}}_{t-\rho-\mu_t} < \underbrace{(h_{\min}+1) = \cdots = (h_{\min}+1)}_{\mu_t} \le \eta_{t+1} \le \cdots \le \eta_q, \tag{51}$$

[illegible] $\mu_t$ [illegible] $\sum_{i=1}^{t} \eta_i + \widetilde{\eta}$ [illegible] $t-\rho$ [illegible]

3. $\mathcal{J}_P \oplus \mathcal{J}_Q$ [illegible] $(P+Q)(\lambda)$ [illegible] $(P+Q)(\lambda)$

[illegible] 5. We noted in subsection 5.5 that the additional assumption $\operatorname{rank}(Q) \le \operatorname{rank}(P)$ is sufficient for considering that the KCF of $(P+Q)(\lambda)$ found in Theorem 6.2 is generic.

[illegible] 6.2. Theorem 5.8 applied to $P$ and $Q$ proves (50) and applied to $P^T$ and $Q^T$ proves (51). Theorem 4.4 proves that for every complex number $\lambda_0$, including the infinite, $\mathcal{S}_{P+Q}(\lambda_0) \ge \mathcal{S}_{P\oplus Q}(\lambda_0)$. To prove that, in fact, $\mathcal{S}_{P+Q}(\lambda_0) = \mathcal{S}_{P\oplus Q}(\lambda_0)$, we will simply show that the direct sum of $\mathcal{J}_P \oplus \mathcal{J}_Q$ plus the column and row singular blocks corresponding to (50) and to (51) is an $m \times n$ pencil. Let us call this direct sum $Z(\lambda)$.

Let the matrix $\mathcal{J}_P$ be $r_1 \times r_1$, and $\mathcal{J}_Q$ be $r_2 \times r_2$. Notice that, in this situation, the following identities hold:

$$\varepsilon + \eta + q + r_1 = m, \quad \varepsilon + p + \eta + r_1 = n, \quad \widetilde{\varepsilon} + \widetilde{\eta} + r_2 = \rho,$$

where $\varepsilon$ $(\eta)$ is the sum of the column (row) minimal indices of $P$. Thus, the number of rows of $Z(\lambda)$ is

$$\begin{aligned}
&[d_{\min}(s-\rho-\gamma_s) + (d_{\min}+1)\gamma_s + \varepsilon_{s+1} + \cdots + \varepsilon_p] \\
&+ [(h_{\min}+1)(t-\rho-\mu_t) + (h_{\min}+2)\mu_t + (\eta_{t+1}+1) + \cdots + (\eta_q+1)] + r_1 + r_2 \\
&= [\varepsilon + \widetilde{\varepsilon}\,] + [\eta + q + \widetilde{\eta} - \rho\,] + r_1 + r_2 = m + \widetilde{\varepsilon} + \widetilde{\eta} + r_2 - \rho = m.
\end{aligned}$$

An analogous computation shows that the number of columns of $Z(\lambda)$ is $n$, and, therefore, that $Z(\lambda)$ is the KCF of $(P+Q)(\lambda)$. ☐

[illegible] 2. Let us apply Theorem 6.2 to the pencil $P(\lambda)$ of the example in subsection 5.4. In that subsection, we considered a perturbation $Q(\lambda)$ with $\rho = 2$ and $\widetilde{\varepsilon} = 1$. Now, let us assume also that $\widetilde{\eta} = 0$ and that $Q$ has a simple eigenvalue $\mu = 1$. The generic column minimal index of $P+Q$ predicted by Theorem 6.2 was computed in subsection 5.4 and is 3. Let us compute the generic row minimal indices of $P+Q$. In this example $\eta_1 = \eta_2 = \eta_3 = 0$. Thus, the number of row minimal indices of $P+Q$ is $q - \rho = 3 - 2 = 1$. Therefore, $h_{\min} = h_3 = 0$ is the generic row minimal index of $P+Q$. The generic KCF of $P+Q$ is

$$\begin{bmatrix} \lambda & 1 & 0 & 0 & 0 \\ 0 & \lambda & 1 & 0 & 0 \\ 0 & 0 & \lambda & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda - 1 \end{bmatrix}.$$

In the case that the only information available on the perturbation $Q(\lambda)$ is its rank, Theorem 4.4 can be combined with Theorem 5.10, and the corresponding counterpart version for row minimal indices, to produce Theorem 6.3, that gives partial information of the KCF of $(P+Q)(\lambda)$.

THEOREM 6.3. [illegible] $P(\lambda)$ [illegible] $Q(\lambda)$ [illegible] $m \times n$ [illegible] $\mathrm{rank}(P) + \mathrm{rank}(Q) < \min\{m, n\}$ [illegible] $\mathrm{rank}(Q) \leq \mathrm{rank}(P)$ [illegible] $\rho \equiv \mathrm{rank}(Q)$ [illegible] $\varepsilon_1 \leq \cdots \leq \varepsilon_p$ [illegible] $\eta_1 \leq \cdots \leq \eta_q$ [illegible] $P$ [illegible] $\mathcal{J}_P$ [illegible] $P$ [illegible]

$$d'_k = \left\lfloor \frac{\sum_{i=1}^{k} \varepsilon_i + \rho}{k - \rho} \right\rfloor \quad [illegible] \quad k = \rho + 1, \ldots, p \; [illegible]$$

$$h'_l = \left\lfloor \frac{\sum_{i=1}^{l} \eta_i + \rho}{l - \rho} \right\rfloor \quad [illegible] \quad l = \rho + 1, \ldots, q.$$

[illegible] $d'_{\min}$ $h'_{\min}$ [illegible] $\{d'_k\}$ $\{h'_l\}$ [illegible] $s'$ $t'$ [illegible] $d'_{s'} = d'_{\min}$ $h'_{t'} = h'_{\min}$ [illegible] $d'_{s'} \geq \varepsilon_{s'}$ $h'_{t'} \geq \eta_{t'}$ [illegible] $Q(\lambda)$ [illegible] $\rho$ $(P+Q)(\lambda)$ [illegible] $p - \rho$ [illegible] $q - \rho$ [illegible]

1. [illegible] $p - s'$ [illegible] $(P+Q)(\lambda)$ [illegible] $\varepsilon_{s'+1} \leq \cdots \leq \varepsilon_p$
2. [illegible] $s' - \rho$ [illegible] $(P+Q)(\lambda)$ $\hat{\varepsilon}_1 \leq \cdots \leq \hat{\varepsilon}_{s'-\rho}$ [illegible] $\varepsilon_{\rho+j} \leq \hat{\varepsilon}_j$ [illegible] $j = 1, \ldots, s' - \rho$ [illegible] $\hat{\varepsilon}_1 \leq d'_{min}$
3. [illegible] $q - t'$ [illegible] $(P+Q)(\lambda)$ [illegible] $\eta_{t'+1} \leq \cdots \leq \eta_q$
4. [illegible] $t' - \rho$ [illegible] $(P+Q)(\lambda)$ $\hat{\eta}_1 \leq \cdots \leq \hat{\eta}_{t'-\rho}$ [illegible] $\eta_{\rho+j} \leq \hat{\eta}_j$ [illegible] $j = 1, \ldots, t' - \rho$ [illegible] $\hat{\eta}_1 \leq h'_{min}$
5. [illegible] $(P+Q)(\lambda)$ [illegible] $\mathcal{J}_P$

**7. Conclusions and open problems.** The results presented in this paper are, as far as we know, the first contribution in the area of [illegible] low rank perturbations of singular matrix pencils, but they do not solve all the problems of this kind.

A first interesting problem is to consider unperturbed pencils $P(\lambda)$ [illegible], i.e., $\mathrm{rank}(P) = \min\{m, n\}$. The full rank square case, $m = n$, corresponds to unperturbed regular pencils. In this case the KCF of $P(\lambda)$ does not have singular blocks, and it is called the Weierstrass canonical form. This problem has been solved in [4]. The full rank rectangular case, $m \neq n$, is an open problem. In this case the KCF of $P(\lambda)$ has only one type of singular blocks: $n-m$ column or right singular blocks if $m < n$, and $m-n$ row or left singular blocks if $m > n$. Generically the same holds for the perturbed pencil $(P+Q)(\lambda)$, but the dimensions of the singular blocks may change. A first important task in this setting is to define the precise meaning of [illegible].

A second open problem is to consider unperturbed pencils $P(\lambda)$ without full rank, but [illegible] (1). For instance, if $P(\lambda)$ is a $100 \times 200$ pencil with $\operatorname{rank}(P) = 98$ and the rank of the perturbations is $\rho \equiv \operatorname{rank}(Q) = 3$, then the perturbations $Q(\lambda)$ are, intuitively, low rank perturbations of $P(\lambda)$. The solution of this kind of problem is naturally connected with the results presented in this work and with the first open problem we have discussed in the previous paragraph. In our specific example, the right decomposition of $Q$ in (7) allows us to write $Q(\lambda) = Q_1(\lambda) + Q_2(\lambda)$, where $\operatorname{rank}(Q_1) = 2$ and $\operatorname{rank}(Q_2) = 1$. Thus, we can split the original perturbation problem, $P+Q = P+Q_1+Q_2$, into two perturbation problems of smaller rank, $P+Q_1$ and $(P+Q_1)+Q_2$. The first one is of the type considered in this work, and in the second one the unperturbed pencil $P+Q_1$ has generically full rank.

A final open problem has to do with the fact that in some situations the information given by the results presented in this paper for the [illegible], i.e., $\operatorname{rank}(P+Q) = \operatorname{rank}(P) + \operatorname{rank}(Q) = \min\{m, n\}$, [illegible] is irrelevant. Notice that in the rectangular case—let us assume $m < n$ without loss of generality—our results say that $P+Q$ does not have row minimal indices, and Theorems 5.8 and 5.10 determine the generic column minimal indices. Additionally, Theorem 4.4 gives information on the regular part of $P+Q$. However, in the square case, although our results are still true, they may produce irrelevant information. Let us illustrate this with two examples. The first example is

$$\underbrace{\begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{P} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \lambda \end{bmatrix}}_{Q} = \underbrace{\begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 2 \\ 0 & 0 & \lambda \end{bmatrix}}_{P+Q}. \tag{52}$$

Notice that $P$ has $\operatorname{rank}(P) = 2$, minimal indices $\varepsilon_1 = 2$ and $\eta_1 = 0$, and no eigenvalues because its rank is 2 for all the values of $\lambda$. The same holds for the dual pencil. Thus, $P$ has no regular part. The pencil $Q$ has $\operatorname{rank}(Q) = 1$, minimal indices $\varepsilon_1 = \varepsilon_2 = 0$ and $\eta_1 = 0, \eta_2 = 1$, and no eigenvalues. However, $P+Q$ has $\operatorname{rank}(P+Q) = 3$; i.e., it is a regular pencil and does not have minimal indices, neither row nor column minimal indices. This is predicted by our theory; see Corollary 3.2. In addition, $P+Q$ has $\mu = 0$ as a triple eigenvalue with only one associated Jordan block. Notice that the information given by Theorem 4.4 is true—$\mathcal{S}_{P+Q}(0) = (3, 0, \dots)$, while $\mathcal{S}_{P \oplus Q}(0) = (0, 0, \dots)$—but irrelevant, because [illegible] $P$ [illegible] $Q$ [illegible] $P+Q$. This first example illustrates a type of perturbation that destroys all the singular information of the unperturbed pencil and creates a regular part in $P+Q$ that does not exist at all in $P$. Therefore the [illegible] of $P+Q$ is created from [illegible] of $P$ and $Q$. Notice that this example is not particular, because once $P$ is fixed and the rank of the perturbations is fixed to be one, it is generic that $\operatorname{rank}(P+Q) = 3$ (see Theorem 3.1), and $P+Q$ has no minimal indices but only a regular part. The second example is the following:

$$\underbrace{\begin{bmatrix} \lambda-1 & 0 \\ 0 & 0 \end{bmatrix}}_{P} + \underbrace{\begin{bmatrix} \lambda & 1+2\lambda \\ 2\lambda & 2+4\lambda \end{bmatrix}}_{Q} = \underbrace{\begin{bmatrix} 2\lambda-1 & 1+2\lambda \\ 2\lambda & 2+4\lambda \end{bmatrix}}_{P+Q}. \tag{53}$$

In this example, the pencil $P$ has $\operatorname{rank}(P)=1$, minimal indices $\varepsilon_1=0$ and $\eta_1=0$, and one simple eigenvalue equal to 1. The pencil $Q$ has $\operatorname{rank}(Q)=1$, minimal indices $\varepsilon_1=1$ and $\eta_1=0$, and no regular part. The pencil $P+Q$ is regular with determinant $\det(P+Q)=2(1+2\lambda)(\lambda-1)$; this means that $P+Q$ has two simple eigenvalues equal to $-1/2$ and 1. Notice that in this case, $\mu=1$ is an eigenvalue of $P$ and also of $P+Q$. This is guaranteed by Theorem 4.4, and it is not a coincidence. But the new eigenvalue appearing in $P+Q$, i.e., $-1/2$, is not related to the regular structure of $P$. In both examples, (52) and (53), it seems difficult to say something generic on the regular part of $P+Q$ beyond Theorem 4.4, except that the new eigenvalues appearing in $P+Q$ will be [illegible] different from those of $P$. However, to find precise conditions for this behavior to hold needs delicate algebraic work and still remains as an open problem.

**Acknowledgments.** The authors thank the editor of this paper, Prof. Ilse Ipsen, and two anonymous referees for many helpful suggestions which have resulted in a great improvement of the presentation.

## REFERENCES

[1] D. L. Boley, *The algebraic structure of pencils and block Toeplitz matrices*, Linear Algebra Appl., 279 (1998), pp. 255–279.

[2] I. de Hoyos, *Points of continuity of the Kronecker canonical form*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 278–300.

[3] F. de Terán and F. M. Dopico, *A note on generic Kronecker orbits of matrix pencils with fixed rank*, SIAM J. Matrix Anal. Appl., submitted, 2006.

[4] F. de Terán, F. M. Dopico, and J. Moro, *Low rank perturbation of Weierstrass structure*, SIAM J. Matrix Anal. Appl., submitted, 2005.

[5] A. Edelman, E. Elmroth, and B. Kågström, *A geometric approach to perturbation theory of matrices and matrix pencils.* Part II. *A stratification-enhanced staircase algorithm*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 667–699.

[6] G. D. Forney, Jr., *Minimal bases of rational vector spaces, with applications to multivariable linear systems*, SIAM J. Control, 13 (1975), pp. 493–520.

[7] M. I. Friswell, U. Prells, and S. D. Garvey, *Low-rank damping modifications and defective systems*, J. Sound Vibration, 279 (2005), pp. 757–774.

[8] F. Gantmacher, *The Theory of Matrices*, AMS Chelsea, Providence, RI, 1998.

[9] I. Gohberg, P. Lancaster, and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[10] L. Hörmander and A. Melin, *A remark on perturbations of compact operators*, Math. Scand., 75 (1994), pp. 255–262.

[11] N. Karcanias, *Minimal bases of matrix pencils: Algebraic Toeplitz structure and geometric properties*, Linear Algebra Appl., 205/206 (1994), pp. 831–868.

[12] H. Langer and B. Najman, *Remarks on the perturbation of analytic matrix functions.* III, Integral Equations Operator Theory, 15 (1992), pp. 796–806.

[13] J. Moro and F. M. Dopico, *Low rank perturbation of Jordan structure*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 495–506.

[14] A. Pokrzywa, *On perturbations and the equivalence orbit of a matrix pencil*, Linear Algebra Appl., 82 (1986), pp. 99–121.

[15] U. Prells, J. E. Mottershead, and M. I. Friswell, *On pole-zero placement by unit-rank modification*, Mechanical Systems and Signal Processing, 17 (2003), pp. 611–633.

[16] S. V. Savchenko, *On a generic change in the spectral properties under perturbation by an operator of rank one*, Mat. Zametki, 74 (2003), pp. 590–602.

[17] S. V. Savchenko, *On the change in the spectral properties of a matrix under a perturbation of a sufficiently low rank*, Funktsional. Anal. i Prilozhen., 38 (2004), pp. 85–88.

[18] R. C. Thompson, *Invariant factors under rank one perturbations*, Canad. J. Math., 32 (1980), pp. 240–245.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 530–553

# STABILITY AND FAST ALGORITHMS OF INCOMPLETE LU FACTORIZATION WITH ZERO-FILL FOR NINE-DIAGONAL MATRICES*

ZHENYUE ZHANG†, MIN FANG†, AND JING WANG†

**Abstract.** Stability and fast algorithms of zero-fill incomplete LU factorization for nonsymmetric nine-diagonal matrices are considered. Based on a convergence analysis of the diagonal sequences in the LU factors of a monotone nine-diagonal matrix, we give necessary and sufficient conditions for stably computing the ILU factorization in terms of an extreme solution to a system of nonlinear equations. Furthermore, the extreme solution is used to construct a fast ILU factorization with much less cost in flops and storage. Basically we use the standard ILU recursions up to a certain point and then replace the remaining terms of the recursive sequences by their limit values that can be determined by the extreme solution. Two strategies, preconditioning the nonlinear system to suit for Newton iterations and estimating the solution to get a good initial for the iterations, are discussed for computing the extreme solution efficiently. We also generalize the stability analysis and propose fast algorithms for nine-diagonal matrices with periodically monotone diagonals or other nonconstant diagonals. Numerical examples show the efficiency of the proposed fast algorithms.

**Key words.** incomplete LU factorization, fast algorithm, stability analysis, nine-diagonal matrix

**AMS subject classifications.** 15A23, 15A06, 65F50

**DOI.** 10.1137/040608702

## 1. Introduction.

Solving a large sparse system of linear equations continues to be a major research task in widespread applications. In general, preconditioning is thought of as a necessary technique for solving a linear system in large scale iteratively. One line of research in preconditioning is based upon incomplete LU factorization (ILU) [2, 12, 16, 17, 21].

The stability of factorization plays an important role in the performance of ILU preconditioning. For symmetric M-matrices or positive definite matrices, ILU factorizations exit and are stable; see [15, 16, 24] for discussions. However, standard ILU factorizations often fail for highly indefinite and nonsymmetric sparse matrices because of the instability [5, 9]. It is known that a small pivot can lead to an unstable ILU process or inaccurate factorization, and the ILU recursions may even break down. In [6], instabilities involved in the incomplete factorizations were discussed in detail. In general, a preprocessing on the coefficient matrix is required before or during factorization to guarantee a stable factorization for the resulting matrix. Such preprocessing strategies include nonsymmetrical permutation and scaling [19, 7, 3], or diagonal compensation [2, 18].

Stability of ILU is generally matrix-dependent. For simple two-dimensional partial differential equations (PDE), five-point difference operators are commonly involved and lead to five-diagonal matrices. In [9], stability analysis of ILU with zero fill-in was given for five-diagonal matrices with constant diagonals that are discretized

---

[*] Received by the editors May 20, 2004; accepted for publication (in revised form) by D. B. Szyld September 25, 2006; published electronically April 20, 2007. This work was supported in part by the Special Funds for Major State Basic Research Projects (project G19990328), NSFC (project 60372033), and foundation by the National Key Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics.

http://www.siam.org/journals/simax/29-2/60870.html

[†] Department of Mathematics, Zhejiang University, Yuquan Campus, Hangzhou, 310027, People's Republic of China (zyzhang@zju.edu.cn, svdfang@hotmail.com, wroaring@sohu.com).

from a simple two-dimensional non-self-adjoint elliptic problem. The stability was characterized by a positive root of a quadratic equation. This analysis was extended to cover the zero fill-in relaxed ILU denoted by RILU($\omega$) in [10] and the 1-level of fill-in RILU($1, \omega$) in [11] for constant five-diagonal matrices. For some PDE such as Fokker–Planck equations [14], cross-derivative terms play an important role. In that case, a nine-point difference operator is more stable and has better convergence than a five-point difference method [20]. Some discussions for preconditioning nine-point approximation to a convection-diffusion equation were given in [4].

This paper deals with a class of nonsymmetric nine-diagonal matrices that are generally obtained by nine-point difference approximations for two-dimensional PDE. We focus our attention on both the stability and related fast algorithms for (approximate) ILU factorization of a nine-diagonal matrix. For a matrix with infinite size, the stability of ILU of a principal submatrix clearly depends on its matrix size. By [illegible], we mean that the ILU of the principal submatrix is still stable when its matrix size increases infinitely. This makes sense since the number of grids of the considered domain may be increased to achieve a higher accuracy, though the matrix size is always finite in practice. Interestingly, the uniform stability is associated with the convergence of nonlinear recursive scheme of the zero-fill ILU factorization. Indeed, a convergence analysis of the nonlinear recursions touches on two important issues: conditions for uniformly stable ILU and fast algorithms for (approximate) ILU factorizations. These two issues are closely related to each other. In fact, if the limit values can be predetermined, and the iterations have achieved to their limits within an acceptable accuracy, then the recursive process can be terminated duly and the remaining terms of the recursive sequences can be simply replaced by their limit values. This idea leads to a fast ILU factorization with much less cost in flops and storage.

We will start our discussion on the uniform stability of ILU of diagonally monotone matrices (nine-diagonal matrices with monotone diagonal sequences in each diagonal). We will give a detailed analysis of the equivalence between uniform stability and recursive convergence of ILU. The convergence analysis of ILU recursions for a diagonally monotone matrix leads to a nonlinear system of equations with multiple variables. We will show that the uniform stability of ILU is equivalent to the existence of positive solutions to the nonlinear equations. Furthermore, an [illegible] determines the limit values of the ILU recursion sequences. For numerical computation, we will give some discussions related to Newton iterations for solving the nonlinear equations: estimation of the extreme solution and preconditioning the nonlinear equations. An important property is that the different sequences converge at [illegible] convergence rates. This gives an easy way to check the convergence of ILU recursions.

The discussion for uniformly stable ILU will be generalized to nine-diagonal matrices with periodically monotone matrices, which also yields a fast ILU factorization in block form. For a matrix which has nonconstant diagonals, an approximate ILU factorization can also be constructed by applying the fast algorithm to a diagonally monotone or periodic matrix which is an approximation to this matrix. Our stability analysis is also true for the zero-fill relaxed ILU (RILU) [2] since the added terms in RILU do not affect the convergence analysis. It can be generalized to RILU with level of fill-in. For simplicity, we focus our discussion on the RILU factorization. The question of selecting the relaxation parameter of RILU is important in applications, but that is beyond the scope of this paper.

The rest of this paper is organized as follows. In section 2, we give a detailed analysis to show the equivalence of uniform stability of RILU factorization and con-

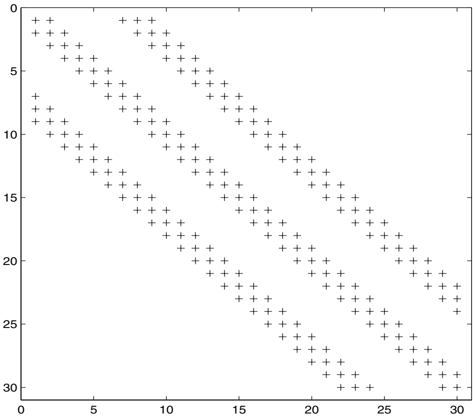


FIG. 1. *The structure of a nine-diagonal matrix of order $N = 30$ with $m = 7$.*

vergence of RILU recursions for diagonally monotone matrices. Some computational issues related to solving the resulting nonlinear system are discussed in section 3. We expand the discussions to more general nine-diagonal matrices such as diagonally periodic matrices in section 4. In section 5, we give fast algorithms for computing the RILU factorization. Some numerical results are reported in section 6.

## 2. The uniform stability of RILU for diagonally monotone matrices.

When a nine-point difference method is applied to a linear non-self-adjoint PDE, the resulting coefficient matrix is a nine-diagonal matrix with the main diagonal and the first, $(m-1)$th, $m$th, and $(m+1)$th upper and lower diagonal lines, where $m \geq 2$ depends on the grid size. Figure 1 illustrates the structure of a nine-diagonal matrix. For simplicity, we write the entries $a_{ij}$ of a nine-diagonal matrix $A = (a_{ij})$ as

$$\begin{array}{lll} a_{i,i} = a_i, & a_{i,i+1} = -b_i, & a_{i+1,i} = -\bar{b}_i, \\ a_{i,i+m-1} = -c_i, & a_{i,i+m} = -d_i, & a_{i,i+m+1} = -e_i, \\ a_{i+m-1,i} = -\bar{c}_i, & a_{i+m,i} = -\bar{d}_i, & a_{i+m+1,i} = -\bar{e}_i, \end{array}$$

and denote the nine-diagonal matrix by

$$A = \{-\bar{e}_i, -\bar{d}_i, -\bar{c}_i, -\bar{b}_i, a_i, -b_i, -c_i, -d_i, -e_i\}.$$

For an elliptic PDE, we may assume that $a_i > 0$ and the rest are nonnegative.

An incomplete LU factorization of $A$ may be written in the form $A = LD^{-1}U - R$ with residual matrix $R$, where $D$ is a diagonal matrix, $L$ is lower triangular and $U$ upper triangular, and the diagonal vectors of $L$, $U$, and $D$ are equal to each other. In the case of zero fill-in, $L$ and $U$ have the same sparse structures as the lower and upper parts of $A$, respectively. For a nine-diagonal matrix $A$, the RILU factorization with zero level of fill-in produces a lower five-diagonal matrix $L$ and an upper five-diagonal $U$ with

$$L = \{-\bar{e}_i, -\bar{\lambda}_i, -\bar{\gamma}_i, -\bar{\beta}_i, \alpha_i, 0, 0, 0, 0\},$$

$$U = \{0, 0, 0, 0, \alpha_i, -\beta_i, -\gamma_i, -\lambda_i, -e_i\},$$

$$D = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N),$$

where $N$ is the size of $A$. The residual matrix $R = LD^{-1}U - A$ is a five-diagonal matrix with zero entries outside of the five diagonals: the main diagonal and the second and $(m-2)$th upper and lower diagonals. It is easy to verify that the RILU factorization can be executed by the following nonlinear recursions:

$$\beta_i = b_i + \frac{\bar\gamma_{i-m+1}\lambda_{i-m+1}}{\alpha_{i-m+1}} + \frac{e_{i-m}\bar\lambda_{i-m}}{\alpha_{i-m}}, \quad \bar\beta_i = \bar b_i + \frac{\gamma_{i-m+1}\bar\lambda_{i-m+1}}{\alpha_{i-m+1}} + \frac{\bar e_{i-m}\lambda_{i-m}}{\alpha_{i-m}}, \tag{2.1}$$

$$\gamma_i = c_i + \lambda_{i-1}\frac{\bar\beta_{i-1}}{\alpha_{i-1}}, \quad \bar\gamma_i = \bar c_i + \bar\lambda_{i-1}\frac{\beta_{i-1}}{\alpha_{i-1}}, \tag{2.2}$$

$$\lambda_i = d_i + e_{i-1}\frac{\bar\beta_{i-1}}{\alpha_{i-1}}, \quad \bar\lambda_i = \bar d_i + \bar e_{i-1}\frac{\beta_{i-1}}{\alpha_{i-1}}, \tag{2.3}$$

$$\begin{aligned}\alpha_i = a_i &- \frac{\big(\beta_{i-1} + \omega\gamma_{i-1}\big)\bar\beta_{i-1}}{\alpha_{i-1}} - \frac{\big(\gamma_{i-m+1} + \omega(e_{i-m+1} + \beta_{i-m+1})\big)\bar\gamma_{i-m+1}}{\alpha_{i-m+1}} \\ &- \frac{\lambda_{i-m}\bar\lambda_{i-m}}{\alpha_{i-m}} - \frac{\big(e_{i-m-1} + \omega\gamma_{i-m-1}\big)\bar e_{i-m-1}}{\alpha_{i-m-1}}.\end{aligned} \tag{2.4}$$

Here we assume $\alpha_1 = a_1$ and

$$\beta_i = \bar\beta_i = \lambda_i = \bar\lambda_i = \gamma_i = \bar\gamma_i = 0$$

if $i \le 0$. Note that if $A$ has a finite size $N$, the terms $e_{N-m}$, $\bar e_{N-m}$, $e_{N-m+1}$, $\bar e_{N-m+1}$, $\gamma_i$, $\bar\gamma_i$ $(i \ge N-m+2)$, $\lambda_i$, $\bar\lambda_i$ $(i \ge N-m+1)$ are not involved in the RILU recursions because they are not in the matrix. So we set $\gamma_i = \bar\gamma_i = 0$ for $i \ge N-m+2$ and $\lambda_i = \bar\lambda_i = 0$ for $i \ge N-m+1$ in (2.4); see Algorithm RILU for details.

The RILU recursions result in the seven sequences $\{\alpha_i\}$, $\{\beta_i\}$, $\{\bar\beta_i\}$, $\{\lambda_i\}$, $\{\bar\lambda_i\}$, $\{\gamma_i\}$, and $\{\bar\gamma_i\}$. For convenience, we call $\{a_i\}$, $\{b_i\}$, $\{c_i\}$, $\{d_i\}$, $\{e_i\}$, $\{\bar b_i\}$, $\{\bar c_i\}$, $\{\bar d_i\}$, and $\{\bar e_i\}$ the input sequences and $\{\alpha_i\}$, $\{\beta_i\}$, $\{\bar\beta_i\}$, $\{\lambda_i\}$, $\{\bar\lambda_i\}$, $\{\gamma_i\}$, and $\{\bar\gamma_i\}$ the output sequences. That ILU of $A$ is uniformly stable implies that $\{\alpha_i\}$ has a positive lower bound. By the recursive formulae (2.2)–(2.4), we have the following obvious observations.

Lemma 2.1. [illegible] $\{\alpha_i\}$ [illegible]

$$\alpha_i \le a_i,\ \beta_i \ge b_i,\ \bar\beta_i \ge \bar b_i,\ \gamma_i \ge c_i,\ \bar\gamma_i \ge \bar c_i,\ \lambda_i \ge d_i,\ \bar\lambda_i \ge \bar d_i. \tag{2.5}$$

[illegible] $\{a_i\}$ [illegible]

[illegible]. Since $\alpha_i > 0$ for all $i$ and all the inputs are nonnegative, by the recursions (2.2)–(2.3), it is obviously true that the outputs except for $\{\alpha_i\}$ are nonnegative at least. So the first assertion is true by (2.2)–(2.4). The second assertion follows from (2.4). In fact, by the assumed properties of the input sequences and (2.5), $\{\alpha_i\}$ is bounded above while the other output sequences have positive lower bounds. Notice that all the sequences are nonnegative. We have by (2.4) that

$$0 < \alpha_i \le a_i - \frac{\beta_{i-1}\bar\beta_{i-1}}{\alpha_{i-1}}, \quad a_i - \frac{\gamma_{i-m+1}\bar\gamma_{i-m+1}}{\alpha_{i-m+1}}, \quad a_i - \frac{\lambda_{i-m}\bar\lambda_{i-m}}{\alpha_{i-m}}.$$

Hence $\beta_i\bar\beta_i \le a_{i+1}a_i$, $\gamma_i\bar\gamma_i \le a_{m+i-1}a_i$, and $\lambda_i\bar\lambda_i \le a_{m+i}a_i$ since $\alpha_i \le a_i$. It follows immediately that $\{\beta_i\}$, $\{\bar\beta_i\}$, $\{\gamma_i\}$, $\{\bar\gamma_i\}$, $\{\lambda_i\}$, and $\{\bar\lambda_i\}$ are bounded above since

Algorithm RILU (relaxed incomplete LU factorization).

```
1. Initialization.
     $\alpha_1 = a_1,$
     $\gamma_1 = c_1,\ \lambda_1 = d_1,\ \beta_1 = b_1,\ t_1 = b_1/a_1,$      $\bar\gamma_1 = \bar c_1,\ \bar\lambda_1 = \bar d_1,\ \bar\beta_1 = \bar b_1,\ \bar t_1 = \bar b_1/a_1,$
     $e_{N-m} = 0,$                                                         $\bar e_{N-m} = 0.$
2. For $i = 2, \dots, N-1$
     if $i < m$
       $\beta_i = b_i,$                                                     $\bar\beta_i = \bar b_i,$
       $\alpha_i = a_i - (\beta_{i-1} + \omega\gamma_{i-1})\bar t_{i-1},$
     else if $i = m$
       $u_1 = \frac{\gamma_1}{\alpha_1},$                                   $\bar u_1 = \frac{\bar\gamma_1}{\alpha_1},$
       $\beta_i = b_i + \lambda_1\bar u_1,$                                 $\bar\beta_i = \bar b_i + \bar\lambda_1 u_1,$
       $\alpha_i = a_i - (\beta_{i-1} + \omega\gamma_{i-1})\bar t_{i-1} - (\gamma_1 + \omega(e_1 + \beta_1))\bar u_1,$
     else if $i = m+1$
       $u_2 = \frac{\gamma_2}{\alpha_2},\ v_1 = \frac{\lambda_1}{\alpha_1},$           $\bar u_2 = \frac{\bar\gamma_2}{\alpha_2},\ \bar v_1 = \frac{\bar\lambda_1}{\alpha_1},$
       $\beta_i = b_i + \lambda_2\bar u_2 + e_1\bar v_1,$                   $\bar\beta_i = \bar b_i + \bar\lambda_2 u_2 + \bar e_1 v_1,$
       $\alpha_i = a_i - \lambda_1\bar v_1 - (\beta_{i-1} + \omega\gamma_{i-1})\bar t_{i-1} - (\gamma_2 + \omega(e_2 + \beta_2))\bar u_2,$
     else
       $u_{i-m+1} = \frac{\gamma_{i-m+1}}{\alpha_{i-m+1}},\ v_{i-m} = \frac{\lambda_{i-m}}{\alpha_{i-m}},$     $\bar u_{i-m+1} = \frac{\bar\gamma_{i-m+1}}{\alpha_{i-m+1}},\ \bar v_{i-m} = \frac{\bar\lambda_{i-m}}{\alpha_{i-m}},$
       $\beta_i = b_i + \lambda_{i-m+1}\bar u_{i-m+1} + e_{i-m}\bar v_{i-m},$     $\bar\beta_i = \bar b_i + \bar\lambda_{i-m+1}u_{i-m+1} + \bar e_{i-m}v_{i-m},$
       $\alpha_i = a_i - (\beta_{i-1} + \omega\gamma_{i-1})\bar t_{i-1} - \lambda_{i-m}\bar v_{i-m}$
            $-(\gamma_{i-m+1} + \omega(e_{i-m+1} + \beta_{i-m+1}))\bar u_{i-m+1} - (e_{i-m-1} + \omega\gamma_{i-m-1})\frac{\bar e_{i-m-1}}{\alpha_{i-m-1}},$
     end if
     if $i \le N-m$
       $\gamma_i = c_i + \lambda_{i-1}\bar t_{i-1},\ \lambda_i = d_i + e_{i-1}\bar t_{i-1},$    $\bar\gamma_i = \bar c_i + \bar\lambda_{i-1}t_{i-1},\ \bar\lambda_i = \bar d_i + \bar e_{i-1}t_{i-1},$
     else if $i = N-m+1$
       $\gamma_i = c_i + \lambda_{i-1}\bar t_{i-1},\ \lambda_i = 0,$                  $\bar\gamma_i = \bar c_i + \bar\lambda_{i-1}t_{i-1},\ \bar\lambda_i = 0,$
     else
       $\gamma_i = 0,\ \lambda_i = 0,$                                      $\bar\gamma_i = 0,\ \bar\lambda_i = 0,$
     end if
     $t_i = \beta_i/\alpha_i,$                                              $\bar t_i = \bar\beta_i/\alpha_i.$
3. $\alpha_N = a_N - \lambda_{N-m}\bar\lambda_{N-m}/\alpha_{N-m} - \beta_{N-1}\bar t_{N-1}$
       $-\big(\gamma_{N-m+1} + \omega\beta_{N-m+1}\big)\bar\gamma_{N-m+1}/\alpha_{N-m+1}$
       $-\big(e_{N-m-1} + \omega\gamma_{N-m-1}\big)\bar e_{N-m-1}/\alpha_{N-m-1}.$
```

they have positive lower bounds. Finally, $\{\alpha_i\}$ has a positive bound because $\alpha_i > \beta_i\bar\beta_i/a_{i+1}$. $\square$

We say $A$ is [illegible] if the input sequences are positive. We also say $A$ is [illegible] if $\{a_i\}$ is decreasing and the other input sequences are increasing. Furthermore, $A$ is said to be [illegible] if its diagonal sequences are convergent. For a diagonally positive and monotone $A$, if $\{\alpha_i\}$ is positive, we can conclude that the outputs are monotone; i.e., $\{\alpha_i\}$ is decreasing and the other output sequences are increasing. This assertion can be easily verified due to the initial monotonicity

$$\alpha_2 \le \alpha_1,\ \beta_2 \ge \beta_1,\ \gamma_2 \ge \gamma_1,\ \lambda_2 \ge \lambda_1,\ \bar\beta_2 \ge \bar\beta_1,\ \bar\gamma_2 \ge \bar\gamma_1,\ \bar\lambda_2 \ge \bar\lambda_1,$$

and the monotone recursions (2.2)–(2.4) of the respective terms. Thus, Lemma 2.1 ensures the convergence of the output sequences. Notice that, by the first assertion of Lemma 2.1, the input sequences are also bounded and hence convergent. Therefore the following theorem is true.

THEOREM 2.2. $A$ $A$ $A$ $\{\alpha_i\}$ $\{\beta_i\}$ $\{\bar\beta_i\}$ $\{\lambda_i\}$ $\{\bar\lambda_i\}$ $\{\gamma_i\}$ $\{\bar\gamma_i\}$

As a necessary condition for uniformly stable RILU, all the input sequences are assumed to be convergent. We denote the limit values by $a$, $b$, $c$, $d$, $\bar a$, $\bar b$, $\bar c$, and $\bar d$, respectively. The limits of the output sequences are denoted by $\alpha^*$, $\beta^*$, $\gamma^*$, $\lambda^*$, $\bar\beta^*$, $\bar\gamma^*$, and $\bar\lambda^*$. By (2.2)–(2.4), we obviously have

$$\beta^* = b + \frac{\lambda^*\bar\gamma^* + e\bar\lambda^*}{\alpha^*}, \qquad \bar\beta^* = \bar b + \frac{\bar\lambda^*\gamma^* + \bar e\lambda^*}{\alpha^*}, \tag{2.6}$$

$$\gamma^* = c + \frac{\lambda^*\bar\beta^*}{\alpha^*}, \qquad \bar\gamma^* = \bar c + \frac{\bar\lambda^*\beta^*}{\alpha^*}, \tag{2.7}$$

$$\lambda^* = d + \frac{e\bar\beta^*}{\alpha^*}, \qquad \bar\lambda^* = \bar d + \frac{\bar e\beta^*}{\alpha^*}, \tag{2.8}$$

and

$$\alpha^* = a - \frac{1}{\alpha^*}\Big(\beta^*\bar\beta^* + \gamma^*\bar\gamma^* + \lambda^*\bar\lambda^* + e\bar e + \omega\big((\bar e + \bar\beta^*)\gamma^* + (e + \beta^*)\bar\gamma^*\big)\Big). \tag{2.9}$$

We will modify the equalities (2.6)–(2.9) equivalently and yield a compact form by introducing $t^* = \beta^*/\alpha^*$ and $\bar t^* = \bar\beta^*/\alpha^*$, the limit values of sequences $\{t_i\}$ and $\{\bar t_i\}$ with $t_i = \beta_i/\alpha_i$ and $\bar t_i = \bar\beta_i/\alpha_i$. To this end, we substitute (2.7) and (2.8) into (2.6) and (2.9). By simple calculations, it is easy to verify that $(\alpha^*, t^*, \bar t^*)$ is a positive solution to the following nonlinear system with variables $\alpha$, $t$, and $\bar t$:

$$(b - t\alpha)\alpha + \lambda\bar\gamma + e\bar\lambda = 0, \tag{2.10}$$

$$(\bar b - \bar t\alpha)\alpha + \bar\lambda\gamma + \bar e\lambda = 0, \tag{2.11}$$

$$(\alpha - a)\alpha + \beta\bar\beta + \gamma\bar\gamma + \lambda\bar\lambda + e\bar e + \omega\big((\bar e + \bar\beta)\gamma + (e + \beta)\bar\gamma\big) = 0, \tag{2.12}$$

where $\lambda = \lambda(\bar t)$, $\gamma = \gamma(\bar t)$, $\bar\lambda = \bar\lambda(t)$, $\bar\gamma = \bar\gamma(t)$, $\beta = \beta(\alpha, t)$, and $\bar\beta = \bar\beta(\alpha, \bar t)$ are simply defined by

$$\lambda = d + e\bar t, \quad \gamma = c + \lambda\bar t, \quad \bar\lambda = \bar d + \bar e t, \quad \bar\gamma = \bar c + \bar\lambda t, \quad \beta = t\alpha, \quad \bar\beta = \bar t\alpha. \tag{2.13}$$

Conversely, if the nonlinear system has a positive solution, the RILU of $A$ is also uniformly stable.

THEOREM 2.3. $A$ $A$ (2.10)–(2.12) $(\alpha, t, \bar t)$

The necessity has been proven. To show the sufficiency, assume that the nonlinear equations (2.10)–(2.12) have a positive solution $(\alpha, t, \bar t)$. It follows from (2.13) that $\beta$, $\bar\beta$, $\lambda$, $\bar\lambda$, $\gamma$, and $\bar\gamma$ are positive. Thus the first terms in each of the three equalities (2.10)–(2.12) must be negative. Hence

$$\beta > b, \quad \bar\beta > \bar b, \quad \alpha < a. \tag{2.14}$$

We use the inequalities above to prove the monotonicity of the outputs

$$\begin{cases} \alpha_{i-1} \ge \alpha_i > \alpha, \\ 0 \le \beta_{i-1} \le \beta_i < \beta, \quad 0 \le \bar\beta_{i-1} \le \bar\beta_i < \bar\beta, \\ 0 \le \gamma_{i-1} \le \gamma_i < \gamma, \quad 0 \le \bar\gamma_{i-1} \le \bar\gamma_i < \bar\gamma, \\ 0 \le \lambda_{i-1} \le \lambda_i < \lambda, \quad 0 \le \bar\lambda_{i-1} \le \bar\lambda_i < \bar\lambda \end{cases} \tag{2.15}$$

by induction. From (2.2)–(2.4), (2.9), and (2.14), we see that

$$\beta_2 = \beta_1 = b < \beta, \quad \bar\beta_2 = \bar\beta_1 = \bar b < \bar\beta, \quad \alpha_1 \ge \alpha_2 = a - \frac{\beta_1\bar\beta_1}{a} > a - \frac{\beta\bar\beta}{\alpha} > \alpha$$

and

$$\lambda_1 \le \lambda_2 = d + e\frac{\bar\beta_1}{\alpha_1} < d + e\frac{\bar\beta}{\alpha} = \lambda, \quad \bar\lambda_1 \le \bar\lambda_2 = \bar d + e\frac{\beta_1}{\alpha_1} < \bar d + e\frac{\beta}{\alpha} = \bar\lambda,$$

$$\gamma_1 \le \gamma_2 = c + \lambda_1\frac{\beta_1}{\alpha_1} < c + \lambda\frac{\beta}{\alpha} = \gamma, \quad \bar\gamma_1 \le \bar\gamma_2 = \bar c + \bar\lambda_1\frac{\beta_1}{\alpha_1} < \bar c + \bar\lambda\frac{\beta}{\alpha} = \bar\gamma.$$

Hence (2.15) is true for $i = 2$.

Assume (2.15) holds for all $i \le k$. Then

$$\lambda_k = d + e\frac{\bar\beta_{k-1}}{\alpha_{k-1}} \le d + e\frac{\bar\beta_k}{\alpha_k} = \lambda_{k+1} \quad \text{and} \quad \lambda_{k+1} = d + e\frac{\bar\beta_k}{\alpha_k} < d + e\frac{\bar\beta}{\alpha} = \lambda.$$

Similarly we have

$$\bar\lambda_k \le \bar\lambda_{k+1} < \bar\lambda, \quad \beta_k \le \beta_{k+1} < \beta, \quad \bar\beta_k \le \bar\beta_{k+1} < \bar\beta,$$

$$\gamma_k \le \gamma_{k+1} < \gamma, \quad \bar\gamma_k \le \bar\gamma_{k+1} < \bar\gamma, \quad \alpha_k < \alpha_{k+1}.$$

It follows from (2.4) that

$$\alpha_{k+1} > a - \frac{1}{\alpha}\Big(\bar\beta\beta + \bar\gamma\gamma + \bar\lambda\lambda + \bar e e - \omega(\bar e\gamma + e\bar\gamma + \bar\gamma\beta + \gamma\bar\beta)\Big) = \alpha.$$

Thus (2.15) also holds for $i = k+1$. We conclude that $\alpha_i > \alpha > 0$ holds for all $i$; i.e., the RILU factorization is uniformly stable. □

The nonlinear system (2.10)–(2.12) may have multiple positive solutions. However, the solution $(\alpha^*, t^*, \bar t^*)$ has the following extreme properties: for any positive solution $(\alpha,\ t,\ \bar t)$ of (2.10)–(2.12),

$$\alpha^* \ge \alpha, \quad t^* \le t, \quad \bar t^* \le \bar t. \tag{2.16}$$

From the proof in the sufficiency part above, $\alpha_i \ge \alpha$ for all $i$ implies $\alpha^* \ge \alpha$. The last two inequalities follow from $t^*\alpha^* = \beta^* = t\alpha$ and $\bar t^*\alpha^* = \bar\beta^* = \bar t\alpha$. Obviously, $(\alpha^*, t^*, \bar t^*)$ is unique. We call it an [illegible] solution of (2.10)–(2.12).

Clearly, $\{\alpha_i\}$, $\{t_i\}$, and $\{\bar t_i\}$ determine the convergence behaviors of the other output sequences. The following theorem shows that all the output sequences converge at equivalent rates. This is an important property for constructing a fast RILU factorization which will be discussed in the next section.

THEOREM 2.4. [illegible] $A$ [illegible] $A$ [illegible]

(a) $$\frac{\bar\gamma^*\lambda^* + e\bar\lambda^*}{(\alpha^*)^2\alpha_1} \le \frac{t^* - t_i}{\alpha_i - \alpha^*} \le \frac{1}{\bar\beta^*}, \quad \frac{\gamma^*\bar\lambda^* + \bar e\lambda^*}{(\alpha^*)^2\alpha_1} \le \frac{\bar t^* - \bar t_i}{\alpha_i - \alpha^*} \le \frac{1}{\beta^*},$$

(b) $$\frac{\bar\gamma^*\lambda^* + e\bar\lambda^*}{\alpha^*\alpha_1} \le \frac{\beta^* - \beta_i}{\alpha_i - \alpha^*} \le \frac{\alpha^*}{\bar\beta^*}, \quad \frac{\gamma^*\bar\lambda^* + \bar e\lambda^*}{\alpha^*\alpha_1} \le \frac{\bar\beta^* - \bar\beta_i}{\alpha_i - \alpha^*} \le \frac{\alpha^*}{\beta^*},$$

(c) $$\lambda^* - \lambda_{i+1} = e(\bar t^* - \bar t_i), \quad \bar\lambda^* - \bar\lambda_{i+1} = \bar e(t^* - t_i),$$

(d) $$\frac{\lambda^*(\bar\gamma^*\lambda^* + e\bar\lambda^*)}{(\alpha^*)^2\alpha_1} \le \frac{\gamma^* - \gamma_{i+2}}{\alpha_i - \alpha^*} \le \frac{e\bar t^* + \lambda^*}{\beta^*}, \quad \frac{\bar\lambda^*(\gamma^*\bar\lambda^* + \bar e\lambda^*)}{(\alpha^*)^2\alpha_1} \le \frac{\bar\gamma^* - \bar\gamma_{i+2}}{\alpha_i - \alpha^*} \le \frac{\bar e t^* + \bar\lambda^*}{\bar\beta^*}.$$

[illegible]. The monotonicity of the sequences will be repeatedly used without comment. Substituting all terms in (2.4) except for the second one by their limit values, we have

$$\alpha_i \geq a^* - \bar{\beta}^* t_{i-1} - \frac{\omega\gamma^*\bar{\beta}^*}{\alpha^*} - \frac{\big(\gamma^* + \omega(e+\beta^*)\big)\bar{\gamma}^*}{\alpha^*} - \frac{\lambda^*\bar{\lambda}^*}{\alpha^*} - \frac{\big(e+\omega\gamma^*\big)\bar{e}}{\alpha^*} = \alpha^* + \bar{\beta}^*(t^* - t_{i-1}).$$

The equality on the right follows from (2.9). So $t^* - t_i \leq t^* - t_{i-1} \leq \frac{\alpha_i - \alpha^*}{\bar{\beta}^*}$. Again, by $\alpha_i \geq \alpha^*$, we get

$$\beta^* - \beta_i = \alpha^* t^* - \alpha_i t_i \leq \alpha^*(t^* - t_i) \leq \frac{\alpha^*}{\bar{\beta}^*}(\alpha_i - \alpha^*).$$

On the other hand, by (2.2) and (2.6),

$$\beta^* - \beta_i \geq (\bar{\gamma}^*\lambda^* + e\bar{\lambda}^*)\left(\frac{1}{\alpha^*} - \frac{1}{\alpha_i}\right) = \frac{\bar{\gamma}^*\lambda^* + e\bar{\lambda}^*}{\alpha^*\alpha_i}(\alpha_i - \alpha^*) \geq \frac{\bar{\gamma}^*\lambda^* + e\bar{\lambda}^*}{\alpha^*\alpha_1}(\alpha_i - \alpha^*)$$

and

$$t^* - t_i = \frac{\beta^*}{\alpha^*} - \frac{\beta_i}{\alpha_i} \geq \frac{\beta^* - \beta_i}{\alpha^*} \geq \frac{\bar{\gamma}^*\lambda^* + e\bar{\lambda}^*}{(\alpha^*)^2\alpha_1}(\alpha_i - \alpha^*).$$

Thus, the bounds of $\frac{t^* - t_i}{\alpha_i - \alpha_i^*}$ and $\frac{\beta^* - \beta_i}{\alpha_i - \alpha_i^*}$ in (a) and (b) hold. Similarly, we can prove (a) and (b) for $\frac{\bar{t}^* - \bar{t}_i}{\alpha_i - \alpha_i^*}$ and $\frac{\bar{\beta}^* - \bar{\beta}_i}{\alpha_i - \alpha_i^*}$. The equalities in (c) follow directly from the definition. Finally, the recursions (2.2) and (2.8) give

$$\gamma^* - \gamma_{i+2} = \lambda^*\bar{t}^* - \lambda_{i+1}\bar{t}_{i+1} = (\lambda^* - \lambda_{i+1})\bar{t}^* + \lambda_{i+1}(\bar{t}^* - \bar{t}_{i+1}) = e\bar{t}^*(\bar{t}^* - \bar{t}_i) + \lambda_{i+1}(\bar{t}^* - \bar{t}_{i+1}).$$

By the inequalities $\bar{t}^* - \bar{t}_{i+1} \leq \bar{t}^* - \bar{t}_i$, $d \leq \lambda_{i+1} \leq \lambda^*$, we have

$$\lambda^*(\bar{t}^* - \bar{t}_{i+1}) = (e\bar{t}^* + d)\,(\bar{t}^* - \bar{t}_{i+1}) \leq \gamma^* - \gamma_{i+2} \leq (e\bar{t}^* + \lambda^*)\,(\bar{t}^* - \bar{t}_i).$$

Therefore (d) follows immediately from the bounds given in (a). $\square$

It is convenient to check the convergence of the RILU recursions using the uniform properties shown in Theorem 2.4, meaning that we only have to check the convergence of $\{\alpha_i\}$, provided that the limit $\alpha^*$ can be predetermined.

## 3. Computing the extreme solution.

It is difficult to formulate solutions of the nonlinear system (2.10)–(2.12). The common method is to use Newton iterations, and that is what we are going to do. However, to improve efficiency, the system of equations (2.10)–(2.12) will be preconditioned such that the resulting nonlinear system also has the extreme solution and can be computed by the Newton method applied on the new system. We also give an estimation of the extreme solution to find an initial guess for the Newton iterations.

### 3.1. Estimation of the extreme solution.

The following lower bounds of $\lambda^*$, $\bar{\lambda}^*$, $\gamma^*$, and $\bar{\gamma}^*$ come directly from (2.13):

$$\lambda^* \geq d, \quad \gamma^* \geq c, \quad \bar{\lambda}^* \geq \bar{d}, \quad \bar{\gamma}^* \geq \bar{c}.$$

By (2.6) we have lower bounds

$$\beta^* \geq b + (\bar{c}d + e\bar{d})/\alpha^*, \quad \bar{\beta}^* \geq \bar{b} + (c\bar{d} + \bar{e}d)/\alpha^* \tag{3.1}$$

of $\beta^*$ and $\bar{\beta}^*$. For convenience, we will use the simple lower bounds $b$ and $\bar{b}$ for $\beta^*$ and $\bar{\beta}^*$. Substituting these bounds into (2.12) and by simple calculations, we obtain

$$(\alpha^*)^2 - a\alpha^* + (b\bar{b} + c\bar{c} + d\bar{d} + e\bar{e}) + \omega\big((\bar{e} + \bar{b})c + (e + b)\bar{c}\big) \le 0,$$

an inequality quadratic in $\alpha^*$. It implies the necessary condition

$$a^2 \ge 4\Big(b\bar{b} + c\bar{c} + d\bar{d} + e\bar{e} + \omega(\bar{e}c + \bar{c}b + e\bar{c} + c\bar{b})\Big)$$

for uniformly stable RILU, and $\alpha^*$ is bounded by the two roots,

$$\frac{a - \sqrt{\Delta}}{2} \le \alpha^* \le \frac{a + \sqrt{\Delta}}{2}, \tag{3.2}$$

where $\Delta = a^2 - 4(b\bar{b} + c\bar{c} + d\bar{d} + e\bar{e} + \omega(\bar{e}c + \bar{c}b + e\bar{c} + c\bar{b}))$. Thus, (3.1) and (3.2) yield the lower bounds

$$t^* \ge \frac{2b}{a + \sqrt{\Delta}} + \frac{4(\bar{c}d + e\bar{d})}{(a + \sqrt{\Delta})^2}, \quad \bar{t}^* \ge \frac{2\bar{b}}{a + \sqrt{\Delta}} + \frac{4(c\bar{d} + \bar{e}d)}{(a + \sqrt{\Delta})^2} \tag{3.3}$$

for $t^* = \beta^*/\alpha^*$ and $\bar{t}^* = \bar{\beta}^*/\alpha^*$. We summarize these results in the following theorem.

THEOREM 3.1. [illegible] $A$ [illegible] $A$ [illegible] $(\alpha^*, t^*, \bar{t}^*)$ [illegible] (2.10)–(2.12) [illegible] (3.2) [illegible] (3.3)

**3.2. Preconditioning (2.10)–(2.12) for the extreme solution.** We will precondition (2.10)–(2.12) and establish an equivalent system of two nonlinear equations that involve $t$ and $\bar{t}$ only. It is obvious that (2.10) and (2.11) are equivalent to

$$\begin{aligned} \alpha &= \tfrac{1}{2t}\Big(b + \sqrt{b^2 + 4t(\lambda\bar{\gamma} + e\bar{\lambda})}\Big) \equiv \phi(t, \bar{t}), \\ \alpha &= \tfrac{1}{2\bar{t}}\Big(\bar{b} + \sqrt{\bar{b}^2 + 4\bar{t}(\bar{\lambda}\gamma + \bar{e}\lambda)}\Big) \equiv \bar{\phi}(t, \bar{t}), \end{aligned} \tag{3.4}$$

respectively. This leads to the first equation $\phi(t, \bar{t}) = \bar{\phi}(t, \bar{t})$.

There are many approaches to derive equivalent representations of $\alpha$ in terms of $t$ and $\bar{t}$ to get the second equation. For example, eliminating the square term of $\alpha$ in (2.10) and (2.12) yields

$$\alpha = \frac{t\Big(\gamma\bar{\gamma} + \lambda\bar{\lambda} + e\bar{e} + \omega(\bar{e}\gamma + e\bar{\gamma})\Big) + (1 + t\bar{t})(\lambda\bar{\gamma} + e\bar{\lambda})}{t\Big(a - \omega(\bar{\gamma}t + \gamma\bar{t})\Big) - b(1 + t\bar{t})} \equiv \eta(t, \bar{t}).$$

Likewise, from (2.11) and (2.12) we obtain

$$\alpha = \frac{\bar{t}\Big(\gamma\bar{\gamma} + \lambda\bar{\lambda} + e\bar{e} + \omega(\bar{e}\gamma + e\bar{\gamma})\Big) + (1 + t\bar{t})(\bar{\lambda}\gamma + \bar{e}\lambda)}{\bar{t}\Big(a - \omega(\bar{\gamma}t + \gamma\bar{t})\Big) - \bar{b}(1 + t\bar{t})} \equiv \bar{\eta}(t, \bar{t}).$$

One can prove that (2.10)–(2.12) have positive solutions if and only if

$$\eta(t, \bar{t}) = \phi(t, \bar{t}) = \bar{\phi}(t, \bar{t}) \tag{3.5}$$

have positive solutions and that $(t^*, \bar{t}^*)$ with $t^* = \frac{\beta^*}{\alpha^*}$ and $\bar{t}^* = \frac{\bar{\beta}^*}{\alpha^*}$ is the smallest positive solution of (3.5).

However, we do not want to let $\eta(t, \bar{t}) = \phi(t, \bar{t})$ be the second equation for determining the extreme solution $(t^*, \bar{t}^*)$ because $\eta(t, \bar{t})$ has relative large partial derivatives nearby $(t^*, \bar{t}^*)$. Newton-like iterative methods converge slowly if partial derivatives are large in a neighborhood of a solution. Fortunately, a flatter function is available and can be derived as follows.

Substitute $\beta = t\alpha$ and $\bar{\beta} = \bar{t}\alpha$ into (2.12), and denoting

$$\zeta = \gamma\bar{\gamma} + \lambda\bar{\lambda} + e\bar{e}, +\omega(\bar{e}\gamma + e\bar{\gamma}),$$

we obtain

$$(1 + t\bar{t})\alpha^2 + (\omega(\bar{\gamma}t + \gamma\bar{t}) - a)\alpha + \zeta = 0. \tag{3.6}$$

Then we have $\alpha = \psi_+(t, \bar{t})$, where

$$\psi_+(t, \bar{t}) = \frac{a - \omega(\bar{\gamma}t + \gamma\bar{t}) + \sqrt{(a - \omega(\bar{\gamma}t + \gamma\bar{t}))^2 - 4(1 + t\bar{t})\zeta}}{2(1 + t\bar{t})} \tag{3.7}$$

is the largest root of the quadratic equation (3.6). We will use $\phi(t, \bar{t}) = \psi_+(t, \bar{t})$ as the second equation to determine the extreme solution $(t^*, \bar{t}^*)$, together with $\phi(t, \bar{t}) = \bar{\phi}(t, \bar{t})$. Theorem 3.3 below shows the equivalence between the system of these two equations and the system (2.10)–(2.12). We remark that the smallest root $\psi_-(t, \bar{t})$ of the quadratic equation (3.6) cannot be chosen to compute $(t^*, \bar{t}^*)$, due to its increasing property shown in the following lemma. This lemma is also used in the proof of Theorem 3.3.

Lemma 3.2. [illegible] $\psi_-(t, \bar{t})$ [illegible] (3.6) [illegible] $t > 0$ [illegible] $\bar{t} > 0$

$$\frac{\partial \psi_-(t, \bar{t})}{\partial t} > 0, \quad \frac{\partial \psi_-(t, \bar{t})}{\partial \bar{t}} > 0.$$

[illegible] Let

$$h(t, \bar{t}) = \omega(\bar{\gamma}t + \gamma\bar{t}) - a, \quad g(t, \bar{t}) = \gamma\bar{\gamma} + \lambda\bar{\lambda} + e\bar{e} + \omega(\bar{e}\gamma + e\bar{\gamma}).$$

We have

$$(1 + t\bar{t})\psi_-^2(t, \bar{t}) + h(t, \bar{t})\psi_-(t, \bar{t}) + g(t, \bar{t}) \equiv 0.$$

Taking the partial derivative with respect to $t$ yields

$$(2(1 + t\bar{t})\psi_- + h)\frac{\partial \psi_-}{\partial t} + F(t, \bar{t}) = 0,$$

where $F(t, \bar{t}) = \bar{t}\psi_-^2 + \frac{\partial h}{\partial t}\psi_- + \frac{\partial g}{\partial t} > 0$ for positive $t$ and $\bar{t}$, since $\frac{\partial h}{\partial t} > 0$ and $\frac{\partial g}{\partial t} > 0$. On the other hand,

$$\psi_- < \frac{\psi_- + \psi_+}{2} = -\frac{h}{2(1 + t\bar{t})}.$$

Hence $2(1 + t\bar{t})\psi_- + h < 0$. Therefore, $\frac{\partial \psi_-}{\partial t} = -\frac{F}{2(1+t\bar{t})\psi_- + h} > 0$. Similarly, one can prove that the partial derivative of $\psi_-$ with respective to $\bar{t}$ is also positive. ☐

THEOREM 3.3. [illegible] (2.10)–(2.12) [illegible]

$$\psi_+(t,\bar t) = \phi(t,\bar t) = \bar\phi(t,\bar t) \tag{3.8}$$

[illegible] $(t^*,\bar t^*)$ [illegible] $\{(t_i,\bar t_i)\}$ [illegible] (3.8)

[illegible]. The proof for sufficiency is simple. If (3.8) has a positive solution $(t,\bar t)$, then the value $\alpha = \psi_+(t,\bar t) = \phi(t,\bar t) = \bar\phi(t,\bar t)$ is positive and hence $(\alpha, t, \bar t)$ satisfies (2.10)–(2.12) by (3.4) and (3.7).

Conversely, if (2.10)–(2.12) has positive solutions, then for the extreme solution $(\alpha^*, t^*, \bar t^*)$, $\alpha^* = \phi(t^*,\bar t^*) = \bar\phi(t^*,\bar t^*)$. So $(t^*,\bar t^*)$ solves (3.8) if we can prove $\alpha^* = \psi_+(t^*,\bar t^*)$. To this end, we consider the RILU of the diagonally constant matrix

$$A = \{-\bar e, -\bar d, -\bar c, -\bar b, a, -b, -c, -d, -e\}.$$

By Theorem 2.3, the RILU factorization is uniformly stable and hence, by Theorem 2.2, the output sequences are positive and monotone convergent: $\{\alpha_i\}$ is decreasing and the others are increasing. Therefore,

$$\alpha_i \ge \alpha_{i+1} \ge a - t_i\bar t_i\alpha_i - \frac{\bar\gamma_{i+1}\gamma_{i+1} + \bar\lambda_{i+1}\lambda_{i+1} + \bar e e}{\alpha_i} - \omega\Big(\frac{\bar e\gamma_{i+1} + e\bar\gamma_{i+1}}{\alpha_i} + \bar\gamma_{i+1}t_i + \gamma_{i+1}\bar t_i\Big).$$

Notice that $\lambda_{i+1} = \lambda(\bar t_i)$ and $\bar\lambda_{i+1} = \bar\lambda(t_i)$ as defined in (2.13). We also have $\gamma_{i+1} \ge \gamma(\bar t_i)$ and $\bar\gamma_{i+1} \ge \bar\gamma(t_i)$. Substituting them into the inequality for $\alpha_i$ above yields the quadratic inequality

$$\begin{aligned}(1 + t_i\bar t_i)\alpha_i^2 &+ \Big(\omega\big(\bar\gamma(t_i)t_i + \gamma(\bar t_i)\bar t_i\big) - a\Big)\alpha_i \\ &+ \gamma(\bar t_i)\bar\gamma(t_i) + \lambda(\bar t_i)\bar\lambda(t_i) + e\bar e + \omega\Big(\bar e\gamma(\bar t_i) + e\bar\gamma(t_i)\Big) \ge 0\end{aligned}$$

for $\alpha_i$. Thus, $\alpha_i \le \psi_-(t_i,\bar t_i)$ or $\alpha_i \ge \psi_+(t_i,\bar t_i)$, where $\psi_-(t_i,\bar t_i)$ and $\psi_+(t_i,\bar t_i)$ are the two roots of the quadratic function. Notice that $(\alpha_i, t_i, \bar t_i) \to (\alpha^*, t^*, \bar t^*)$. So $\alpha^*$ is a root of the quadratic equation (3.6) with $t = t^*$ and $\bar t = \bar t^*$, and the limit $\psi_-(t^*,\bar t^*)$ of $\{\psi_-(t_i,\bar t_i)\}$ is the smallest root. We conclude that $\alpha_i \ge \psi_+(t_i,\bar t_i)$ for all $i$ because $\{\alpha_i\}$ is monotonic decreasing and $\{\psi_-(t_i,\bar t_i)\}$ is increasing by Lemma 3.2. Therefore $\alpha^* = \psi_+(t^*,\bar t^*)$, completing the proof. □

Notice that the equation $\phi = \psi_-$ may give unwanted solutions, and $\psi_+$ has a much smaller derivative than $\eta$ in a neighborhood of the smallest solution $(t^*,\bar t^*)$. We illustrate these phenomena by two symmetric examples with different values of $b = \bar b$, $c = \bar c$, $d = \bar d$, $e = \bar e$. Obviously, $t = \bar t$ and $\phi = \bar\phi$, and we simply write $\phi(t,\bar t)$ as $\phi(t)$ and do the same for other functions. In Figure 2, we plot the four curves of $\phi$, $\eta$, $\psi_+$, and $\psi_-$ of two examples. Obviously, the nonlinear equation $\phi = \psi_-$ does not take $t^*$ as a solution. Meanwhile, $\eta = \psi_-$ may have no solutions, as shown in the diagraph on the left, or an unwanted solution, in the diagraph on the right. Compared with $\eta$, $\psi_+$ has a much smaller derivative than $\eta$ nearby the smallest solution. As we mentioned, this property makes the Newton method converge quickly if we use $\phi = \psi_+$ rather than $\phi = \eta$.

We also illustrate the equivalence between the convergence of the Newton iterations and the uniform stability of the RILU factorization. The matrix used in each testing is

$$A = \{-\bar e, -\bar d, -\bar c, -\bar b,\ a, -b, -c, -d, -e\}$$

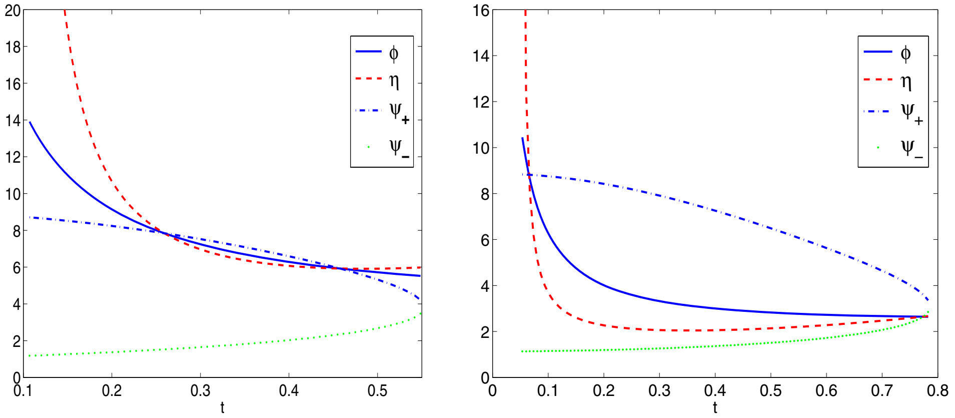


FIG. 2. *Plots of the curves $\phi(t)$, $\eta(t)$, $\psi_+(t)$, and $\psi_-(t)$ for symmetric $A$. The equation $\phi = \psi_-$ gives an unwanted solution. $\psi_+$ is flatter than $\eta$ nearby the smallest solution $t^*$ of $\phi = \psi_+$.*

with fixed $b = 2$, $\bar{b} = 3$, $c = 3$, $\bar{c} = 2$, $d = 2$, $\bar{d} = 3$, $e = 2$, $\bar{e} = 1$ and one of eight values of $a$ from 13.25 to 13.4. Half of the eight values yield uniformly stable ILU, and the other half yield unstable ILU. In each testing, we also compute the extreme solution of (2.10)–(2.12) by Newton iterations, using the lower bounds

$$t_0 = \frac{2b}{a + \sqrt{\Delta}} + \frac{4(\bar{c}d + e\bar{d})}{(a + \sqrt{\Delta})^2}, \quad \bar{t}_0 = \frac{2\bar{b}}{a + \sqrt{\Delta}} + \frac{4(c\bar{d} + \bar{e}d)}{(a + \sqrt{\Delta})^2} \tag{3.9}$$

given by Theorem 3.1 as the initial guess. The convergence of the Newton method conforms to the uniform stability of ILU completely. See Table 1 for the computed results, where "div.," "conv.," "unstab.," and "u.stab." stand for "divergent," "convergent," "unstable," and "uniformly stable," respectively.

TABLE 1
*Consistency between the convergence of Newton iterations and the uniform stability of ILU.*

| $a$ | 13.25 | 13.30 | 13.34 | 13.35 | 13.36 | 13.37 | 13.38 | 13.40 |
|---|---|---|---|---|---|---|---|---|
| Newton | div. | div. | div. | div. | conv. | conv. | conv. | conv. |
| ILU | unstab. | unstab. | unstab. | unstab. | u.stab. | u.stab. | u.stab. | u.stab. |

## 4. Generalization for diagonally variable matrices.

For the general case when $A$ is diagonally variable, it is quite difficult to give a detailed analysis for the uniform stability of RILU factorization. In this section, we will generalize the stability analysis in two cases: (1) nine-diagonal matrices [illegible] by diagonally constant matrices and (2) periodically monotone matrices.

DEFINITION 4.1. [illegible]

$$A = \{-\bar{e}_i, -\bar{d}_i, -\bar{c}_i, -\bar{b}_i, a_i, -b_i, -c_i, -d_i, -e_i\},$$
$$A^* = \{-\bar{e}_i^*, -\bar{d}_i^*, -\bar{c}_i^*, -\bar{b}_i^*, a_i^*, -b_i^*, -c_i^*, -d_i^*, -e_i^*\}$$

[illegible] $A$ [illegible] $A^*$ [illegible] $a_i \geq a_i^*$ [illegible]

$$e_i \leq e_i^*, \quad d_i \leq d_i^*, \quad c_i \leq c_i^*, \quad b_i \leq b_i^*,$$
$$\bar{e}_i \leq \bar{e}_i^*, \quad \bar{d}_i \leq \bar{d}_i^*, \quad \bar{c}_i \leq \bar{c}_i^*, \quad \bar{b}_i \leq \bar{b}_i^*.$$

There are many ways to construct a diagonally constant or monotone matrix that dominates $A$. For example, defining

$$a^* = \inf_k a_k, \quad b^* = \sup_k b_k, \quad c^* = \sup_k c_k, \quad d^* = \sup_k d_k, \quad e^* = \sup_k e_k \tag{4.1}$$

yields a diagonally constant matrix $A^*$, while setting

$$a_i^* = \min_{k\le i} a_k, \quad b_i^* = \max_{k\le i} b_k, \quad c_i^* = \max_{k\le i} c_k, \quad d_i^* = \max_{k\le i} d_k, \quad e_i^* = \max_{k\le i} e_k \tag{4.2}$$

produces a diagonally monotone matrix $A^*$. Clearly, the two matrices defined by (4.1) or (4.2) dominate $A$.

Let $A^* = L^*(D^*)^{-1}U^* - R^*$ be the RILU factorization of a dominant matrix $A^*$ of $A$, where

$$L^* = \{-\bar{e}_i^*, -\bar{\lambda}_i^*, -\bar{\gamma}_i^*, -\bar{\beta}_i^*, \alpha_i^*, 0, 0, 0, 0\}, \quad U^* = \{0, 0, 0, 0, \alpha_i^*, -\beta_i^*, -\gamma_i^*, -\lambda_i^*, -e_i^*\},$$

and $D^* = \mathrm{diag}(\ldots, \alpha_i^*, \ldots)$.

THEOREM 4.2. [illegible] $A$ [illegible] $A^*$ [illegible] $A^*$ [illegible] $A$ [illegible] $L$ $D$ [illegible] $U$ [illegible] $L^*$ $D^*$ [illegible] $U^*$ [illegible]

$$\alpha_i^* \le \alpha_i, \tag{4.3}$$

$$\beta_i^* \ge \beta_i, \quad \gamma_i^* \ge \gamma_i, \quad \lambda_i^* \ge \lambda_i, \quad \bar{\beta}_i^* \ge \bar{\beta}_i, \quad \bar{\gamma}_i^* \ge \bar{\gamma}_i, \quad \bar{\lambda}_i^* \ge \bar{\lambda}_i. \tag{4.4}$$

[illegible] We prove the theorem by induction. Obviously (4.3) and (4.4) hold for $i = 1$ since $\alpha_1 = a_1 \ge a_1^* = \alpha_1^*$ and $\beta_1 = b_1 \le b_1^* = \beta_1^*$.

Assume that (4.3) and (4.4) hold for $i \le k-1$. Since $A$ is dominated by $A^*$, it follows from (2.3) that

$$\lambda_k = d_k + e_k \frac{\beta_{k-1}}{\alpha_{k-1}} \le d_k^* + e_k^* \frac{\beta_{k-1}^*}{\alpha_{k-1}^*} = \lambda_k^*.$$

Similarly, by (2.2)–(2.3), one can prove the other inequalities in (4.4) for $i = k$. Then (4.3) with $i = k$ follows from (2.4) and (4.4) immediately, completing the proof. □

For a diagonally positive, monotone, and convergent nine-diagonal matrix $A$, we denote by

$$A_* = \{-\bar{e}, -\bar{d}, -\bar{c}, -\bar{b}, a, -b, -c, -d, -e\}$$

the diagonally constant matrix consisting of the limit values of the input sequences. Obviously, $A$ is dominated by $A_*$. Using Theorem 2.3, the following equivalence between $A$ and $A_*$ can be proven easily.

THEOREM 4.3. [illegible] $A$ [illegible] $A$ [illegible] $A_*$ [illegible]

**4.1. Block-convergence for periodically monotone matrices.** In general, matrices obtained from two-dimensional PDE by a nine-point difference scheme are not diagonally constant or monotone. Because of the block-tridiagonal form of the resulting matrices, the off-diagonals are periodically zero with period $m$. These periodic zeros can be simply replaced by constants to construct a dominate matrix $A^*$

as in (4.1) or (4.2), if $A$ is diagonally monotone except for the periodic zeros. However, the conditions of uniformly stable RILU of $A$ in terms of the parameters in $A^*$ may be unnecessarily strict. In this subsection, we consider a more general case for nine-diagonal matrices with periodically monotone properties without involving a dominant matrix of $A$.

For a sequence $\{x_i\}$, let

$$x^{(k)} = (x_{km+1}, x_{km+2}, \ldots, x_{km+m})$$

be the $k$th periodic section of $\{x_i\}$ with period $m$, where $m$ is the same as before. Obviously, $\{x^{(k)}\}$ is increasing (convergent) if and only if for any (fixed) $i$, $1 \le i \le m$, the subsequence $\{x_{km+i}\}_{k=1}^{\infty}$ is increasing (convergent).

DEFINITION 4.4. [illegible] $A = \{-\bar{e}_i, -\bar{d}_i, -\bar{c}_i, -\bar{b}_i, a_i, -b_i, -c_i, -d_i, -e_i\}$ [illegible] $m$ [illegible] $\{a^{(k)}\}$ [illegible] $\{b^{(k)}\}$ [illegible]

THEOREM 4.5. [illegible] $A = \{-\bar{e}_i, -\bar{d}_i, -\bar{c}_i, -\bar{b}_i, a_i, -b_i, -c_i, -d_i, -e_i\}$ [illegible] $m$ [illegible] $\{a_i\}$ [illegible] $\{b_i\}$ $\{c_i\}$ $\{d_i\}$ $\{e_i\}$ $\{\bar{b}_i\}$ $\{\bar{c}_i\}$ $\{\bar{d}_i\}$ [illegible] $\{\bar{e}_i\}$ [illegible] $A$ [illegible] $\{\alpha^{(k)}\}$ $\{\beta^{(k)}\}$ $\{\bar{\beta}^{(k)}\}$ $\{\lambda^{(k)}\}$ $\{\bar{\lambda}^{(k)}\}$ $\{\gamma^{(k)}\}$ $\{\bar{\gamma}^{(k)}\}$ [illegible]

[illegible] We first prove that $\{\alpha^{(k)}\}$ is decreasing and $\{\beta^{(k)}\}$ and the others are increasing; i.e.,

$$\begin{aligned}
&\alpha_{i+km} \ge \alpha_{i+(k+1)m}, \\
&\beta_{i+km} \le \beta_{i+(k+1)m}, \quad \lambda_{i+km} \le \lambda_{i+(k+1)m}, \quad \gamma_{i+km} \le \gamma_{i+(k+1)m}, \\
&\bar{\beta}_{i+km} \le \bar{\beta}_{i+(k+1)m}, \quad \bar{\lambda}_{i+km} \le \bar{\lambda}_{i+(k+1)m}, \quad \bar{\gamma}_{i+km} \le \bar{\gamma}_{i+(k+1)m}
\end{aligned} \tag{4.5}$$

for $i = 1, \ldots, m$ and $k \ge 0$.

Consider the initial case when $k = 0$. Since $\alpha_i > 0$ for all $i$, all the output sequences are nonnegative (see Lemma 2.1). It follows immediately that (4.5) holds for $i = 1$. Note that for $i \le m-1$, $\beta_{m+i} \ge b_{m+i} = b_i = \beta_i$ and $\bar{\beta}_{m+i} \ge \bar{\beta}_i$. Assume that (4.5) is true for $i < j$ with fixed $j \le m$. Then for $i = j$,

$$\gamma_{m+i} = c_{m+i} + \lambda_{m+i-1}\frac{\bar{\beta}_{m+i-1}}{\alpha_{m+i-1}} \ge c_i + \lambda_{i-1}\frac{\bar{\beta}_{i-1}}{\alpha_{i-1}} = \gamma_i,$$

and also $\bar{\gamma}_{m+i} \ge \bar{\gamma}_i$, $\lambda_{m+i} \ge \lambda_i$, and $\bar{\lambda}_{m+i} \ge \bar{\lambda}_i$. If $i = j < m$, by (2.4) we have

$$\begin{aligned}
\alpha_{m+i} &= a_{m+i} - \frac{\big(\beta_{m+i-1} + \omega\gamma_{m+i-1}\big)\bar{\beta}_{m+i-1}}{\alpha_{m+i-1}} \\
&\quad - \frac{\big(\gamma_{i+1} + \omega(e_{i+1} + \beta_{i+1})\big)\bar{\gamma}_{i+1}}{\alpha_{i+1}} - \frac{\omega\bar{e}_{i-1}\gamma_{i-1}}{\alpha_{i-1}} \\
&\le a_{m+i} - \frac{\big(\beta_{m+i-1} + \omega\gamma_{m+i-1}\big)\bar{\beta}_{m+i-1}}{\alpha_{m+i-1}} \\
&\le a_i - \frac{\big(\beta_{i-1} + \omega\gamma_{i-1}\big)\bar{\beta}_{i-1}}{\alpha_{i-1}} = \alpha_i.
\end{aligned}$$

If $i = j = m$, we also have

$$\alpha_{m+i} \le a_{m+i} - \frac{\big(\beta_{m+i-1} + \omega\gamma_{m+i-1}\big)\bar{\beta}_{m+i-1}}{\alpha_{m+i-1}} - \frac{\big(\gamma_{i+1} + \omega(e_{i+1} + \beta_{i+1})\big)\bar{\gamma}_{i+1}}{\alpha_{i+1}} = \alpha_i.$$

So (4.5) holds for $i \le m$ when $k = 0$.

Because of the periodicity assumption of $A$, (4.5) holds for all $k \ge 0$ and $i \le m$. For example, if (4.5) is true for $k = j - 1$, then for $k = j$,

$$\begin{aligned}\beta_{km+i} &= b_{km+i} + \frac{\bar{\gamma}_{(k-1)m+i+1}\lambda_{(k-1)m+i+1}}{\alpha_{(k-1)m+i+1}} + \frac{e_{(k-1)m+i}\bar{\lambda}_{(k-1)m+i}}{\alpha_{(k-1)m+i}} \\ &\le b_{(k+1)m+i} + \frac{\bar{\gamma}_{km+i+1}\lambda_{km+i+1}}{\alpha_{km+i+1}} + \frac{e_{km+i}\bar{\lambda}_{km+i}}{\alpha_{km+i}} = \beta_{(k+1)m+i}.\end{aligned}$$

The $m$-block convergence of the output sequences follows from the boundedness of $\{\alpha_i\}$. In fact,

$$0 < \alpha_{i+1} \le a_{i+1} - \frac{\beta_i\bar{\beta}_i}{\alpha_i} \le a_{i+1} - \frac{b\bar{b}}{\alpha_i} \le a_1 - \frac{b\bar{b}}{\alpha_i},$$

giving $\alpha_i > \frac{b\bar{b}}{a_1}$. Similar to Lemma 2.1, using the $m$-block monotonicity in (4.5), we conclude that $\{\beta_i\}$, $\{\bar{\beta}_i\}$, $\{\lambda_i\}$, $\{\bar{\lambda}_i\}$, $\{\gamma_i\}$, and $\{\bar{\gamma}_i\}$ are bounded above. Therefore, all these sequences are $m$-block convergent, and $\{\alpha^{(k)}\}$ converges to a positive vector. ☐

Let $\alpha^* = (\alpha_1^*, \dots, \alpha_m^*)$ be the limit vector of the vector sequence $\{\alpha^{(k)}\}$, and we similarly define $\beta^*$, $\gamma^*$, $\lambda^*$, $\bar{\beta}^*$, $\bar{\gamma}^*$, and $\bar{\lambda}^*$. Also let $A_*$ be the diagonally periodic matrix consisting of these limit vectors. Similar to Theorem 4.3, one can prove that the RILU of a diagonally $m$-monotone matrix $A$ is uniformly stable if and only if the RILU of $A_*$ is uniformly stable.

These limit vectors can be obtained by solving nonlinear equations that are constructed as follows. Let $t_i^* = \beta_i^*/\alpha_i^*$ and $\bar{t}_i^* = \bar{\beta}_i^*/\alpha_i^*$. For $i = 1, \dots, m$, we have

$$\alpha_i^* t_i^* = b_i + \frac{\bar{\gamma}_{i+1}^*\lambda_{i+1}^*}{\alpha_{i+1}^*} + \frac{e_i\bar{\lambda}_i^*}{\alpha_i^*}, \quad \alpha_i^*\bar{t}_i^* = \bar{b}_i + \frac{\gamma_{i+1}^*\bar{\lambda}_{i+1}^*}{\alpha_{i+1}^*} + \frac{\bar{e}_i\lambda_i^*}{\alpha_i^*}, \tag{4.6}$$

$$\gamma_i^* = c_i + \lambda_{i-1}^*\bar{t}_{i-1}^*, \quad \bar{\gamma}_i^* = \bar{c}_i + \bar{\lambda}_{i-1}^* t_{i-1}^*, \tag{4.7}$$

$$\lambda_i^* = d_i + e_{i-1}\bar{t}_{i-1}^*, \quad \bar{\lambda}_i^* = \bar{d}_i + \bar{e}_{i-1} t_{i-1}^*, \tag{4.8}$$

$$\begin{aligned}\alpha_i^* = a_i \quad &- \alpha_{i-1}^* t_{i-1}^*\bar{t}_{i-1}^* - \tfrac{(\gamma_{i+1}^*+\omega e_{i+1})\bar{\gamma}_{i+1}^*}{\alpha_{i+1}^*} - \tfrac{\bar{\lambda}_i^*\lambda_i^*}{\alpha_i^*} \\ &- \tfrac{(e_{i-1}+\omega\gamma_{i-1}^*)\bar{e}_{i-1}}{\alpha_{i-1}^*} - \omega(\bar{t}_{i-1}^*\gamma_{i-1}^* + t_{i+1}^*\bar{\gamma}_{i+1}^*),\end{aligned} \tag{4.9}$$

where $e_0 = e_m$, $e_{m+1} = e_1$, $\bar{e}_0 = \bar{e}_m$, and $\bar{e}_{m+1} = \bar{e}_1$. In the same vein, $\alpha_0^* = \alpha_m^*$ and $\alpha_{m+1}^* = \alpha_1^*$. It is similar for other limit values. Let

$$\lambda_i(t) = d_i + e_{i-1}t, \quad \bar{\lambda}_i(t) = \bar{d}_i + \bar{e}_{i-1}t, \quad \gamma_i(t,s) = c_i + \lambda_{i-1}(s)t, \quad \bar{\gamma}_i(t,s) = \bar{c}_i + \bar{\lambda}_{i-1}(s)t$$

with the periodicity constraints: $\alpha_i = \alpha_j$, $t_i = t_j$, and $\bar{t}_i = \bar{t}_j$, if $|i - j| = m$. Then $\lambda_i^* = \lambda_i(\bar{t}_{i-1}^*)$, $\bar{\lambda}_i^* = \bar{\lambda}_i(t_{i-1}^*)$, $\gamma_i^* = \gamma_i(\bar{t}_{i-1}^*, \bar{t}_{i-2}^*)$, and $\bar{\gamma}_i^* = \bar{\gamma}_i(t_{i-1}^*, t_{i-2}^*)$. Substituting these equalities into (4.6) and (4.9) yields

$$f(\alpha, t, \bar{t}) = 0, \quad \bar{f}(\alpha, t, \bar{t}) = 0, \quad g(\alpha, t, \bar{t}) = 0, \tag{4.10}$$

in the variables $\alpha = (\alpha_1, \dots, \alpha_m)^T$, $t = (t_1, \dots, t_m)^T$, and $\bar{t} = (\bar{t}_1, \dots, \bar{t}_m)^T$, where $f$, $\bar{f}$, and $g$ are $m$-dimensional vector functions with components

$$f_i(\alpha, t, \bar{t}) = b_i - \alpha_i t_i + \frac{\bar{\gamma}_{i+1}(t_i, t_{i-1})\lambda_{i+1}(\bar{t}_i)}{\alpha_{i+1}} + \frac{e_i\bar{\lambda}_i(t_{i-1})}{\alpha_i},$$

$$\begin{aligned}
\bar{f}_i(\alpha, t, \bar{t}) &= \bar{b}_i - \alpha_i \bar{t}_i + \frac{\gamma_{i+1}(\bar{t}_i, \bar{t}_{i-1}) \bar{\lambda}_{i+1}(t_i)}{\alpha_{i+1}} + \frac{\bar{e}_i \lambda_i(\bar{t}_{i-1})}{\alpha_i}, \\
g_i(\alpha, t, \bar{t}) &= \alpha_i - a_i + \alpha_{i-1} t_{i-1} \bar{t}_{i-1} + \frac{\big(\gamma_{i+1}(\bar{t}_i, \bar{t}_{i-1}) + \omega e_{i+1}\big) \bar{\gamma}_{i+1}(t_i, t_{i-1})}{\alpha_{i+1}} \\
&\qquad + \frac{\bar{\lambda}_i(t_{i-1}) \lambda_i(\bar{t}_{i-1})}{\alpha_i} + \frac{\big(e_{i-1} + \omega \gamma_{i-1}(\bar{t}_{i-2}, \bar{t}_{i-3})\big) \bar{e}_{i-1}}{\alpha_{i-1}} \\
&\qquad + \omega \Big( \bar{t}_{i-1} \gamma_{i-1}(\bar{t}_{i-2}, \bar{t}_{i-3}) + t_{i+1} \bar{\gamma}_{i+1}(t_i, t_{i-1}) \Big).
\end{aligned}$$

As shown in the last section, the limit vectors $\alpha^* = (\alpha_1^*, \ldots, \alpha_m^*)$, $t^* = (t_1^*, \ldots, t_m^*)$, and $\bar{t}^* = (\bar{t}_1^*, \ldots, \bar{t}_m^*)$ are the extreme solutions to the nonlinear system (4.10). Newton-like iterations can solve (4.10) if we first run the RILU for several steps to get a good initial guess for the extreme solutions.

**5. Fast algorithms of approximate RILU.** In this section, we give two fast algorithms for computing RILU factorization, one for diagonally monotone matrices and the other for diagonally $m$-monotone matrices. Basically, we first run the standard RILU recursions until the output diagonal sequences are convergent or block-convergent within a given accuracy, and then replace the remaining terms in the recursions by their limit values, except for the last few components of $\{\alpha_i\}$.

**5.1. Fast RILU for diagonally monotone matrices.** For a given diagonally monotone nine-diagonal matrix $A$ of order $N$, we set the parameters in the definitions (3.4) and (3.7) of $\phi(t, \bar{t})$, $\bar{\phi}(t, \bar{t})$, and $\psi_+(t, \bar{t})$ by the last components of the input sequences. For example, set $a = a_N$, $b = b_{N-1}$, $c = c_{N-m+1}$, $d = d_{N-m}$, and $e = e_{N-m-1}$. We solve the nonlinear equations (3.8) for the smallest solution $(t^*, \bar{t}^*)$ by Newton iterations, starting at the initial guess shown in (3.9). (One can also run several steps of RILU recursions first and take the last $(t_i, \bar{t}_i)$ as an initial guess for the Newton iteration.) The stopping criterion is

$$\left\| \begin{pmatrix} t^{(k+1)} - t^{(k)} \\ \bar{t}^{(k+1)} - \bar{t}^{(k)} \end{pmatrix} \right\| < \tau \tag{5.1}$$

with accuracy parameter $\tau$. As soon as $t^*$ and $\bar{t}^*$ are computed, we set $\alpha^* = \psi_+(t^*, \bar{t}^*)$, $\beta^* = \alpha^* t^*$, $\bar{\beta}^* = \alpha^* \bar{t}^*$, and $\gamma^*$, $\bar{\gamma}$, $\lambda^*$, $\bar{\lambda}^*$ as defined in (2.7)–(2.8). We use

$$|\alpha_k - \alpha^*| < \varepsilon \alpha^* \tag{5.2}$$

as the stopping criterion of the RILU recursions with accuracy parameter $\varepsilon$. Because the matrix size is always finite in applications, the terms $\gamma_i$, $\bar{\gamma}_i$ $(i \geq N-m+2)$ and $\lambda_i$, $\bar{\lambda}_i$ $(i \geq N-m+1)$, including $\beta_N$, $\bar{\beta}_N$, and $e_{N-m}$, are not in the matrix. Therefore, for $\max(k+1, N-m+3) \leq i \leq N$, the last several terms $\alpha_i$ are evaluated by

$$\begin{aligned}
\alpha_i \;=\; a_i &- \tfrac{\beta_{i-1} \bar{\beta}_{i-1}}{\alpha_{i-1}} - \tfrac{\big(\gamma_{i-m+1} + \omega(e_{i-m+1} + \beta_{i-m+1})\big) \bar{\gamma}_{i-m+1}}{\alpha_{i-m+1}} \\
&- \tfrac{\lambda_{i-m} \bar{\lambda}_{i-m}}{\alpha_{i-m}} - \tfrac{\big(e_{i-m-1} + \omega \gamma_{i-m-1}\big) \bar{e}_{i-m-1}}{\alpha_{i-m-1}}, \quad i \leq N-2,
\end{aligned} \tag{5.3}$$

$$\begin{aligned}
\alpha_i \;=\; a_i &- \tfrac{\beta_{i-1} \bar{\beta}_{i-1}}{\alpha_{i-1}} - \tfrac{\big(\gamma_{i-m+1} + \omega \beta_{i-m+1}\big) \bar{\gamma}_{i-m+1}}{\alpha_{i-m+1}} \\
&- \tfrac{\lambda_{i-m} \bar{\lambda}_{i-m}}{\alpha_{i-m}} - \tfrac{\big(e_{i-m-1} + \omega \gamma_{i-m-1}\big) \bar{e}_{i-m-1}}{\alpha_{i-m-1}}, \quad i = N-1, N.
\end{aligned} \tag{5.4}$$

Note that if $k \leq N - 2m + 2$, all the terms on the right of (5.3) and (5.4) can be replaced by their limit values.

Now we are ready to present the fast algorithm FRILU for computing RILU factorization of a diagonally monotone matrix.

```
ALGORITHM FRILU-I (FAST RILU FOR DIAGONALLY MONOTONE MATRICES).
Step 0. Initialization.
        (0) Set $a = a_N,\ b = b_{N-1},\ c = c_{N-m+1},\ d = d_{N-m},\ e = e_{N-m-1},$
            $\bar{b} = \bar{b}_{N-1},\ \bar{c} = \bar{c}_{N-m+1},\ \bar{d} = \bar{d}_{N-m},\ \bar{e} = \bar{e}_{N-m-1}.$
        (1) Determine the thresholds $\tau$ for Newton iteration and $\varepsilon$ for RILU.
Step 1. Compute the limits of the output sequences of RILU.
        (2) Solve (3.8) for the smallest positive $(t^*, \bar{t}^*)$ by Newton method with the
            stopping criterion (5.1).
        (3) If the computed $(t^*, \bar{t}^*)$ is not positive (no stable RILU), terminate.
            Otherwise set $\alpha^* = \psi(t^*, \bar{t}^*)$, $\beta^* = \alpha^* t^*$, $\bar{\beta}^* = \alpha^* \bar{t}^*$, and $\gamma^*$, $\bar{\gamma}$, $\lambda^*$, $\bar{\lambda}^*$
            as in (2.7)–(2.8).
Step 2. Partial RILU recursions.
        (4) Run the algorithm RILU until $\alpha_k$ satisfies $|\alpha_k - \alpha^*| < \varepsilon\alpha^*$ or $k = N$.
Step 3. Set the remaining outputs.
        (5) Set $\beta_i = \beta^*,\ \gamma_i = \gamma^*,\ \lambda_i = \lambda^*,\ \bar{\beta}_i = \bar{\beta}^*,\ \bar{\gamma}_i = \bar{\gamma}^*,\ \bar{\lambda}_i = \bar{\lambda}^*$ for $i > k$.
        (6) Set $\alpha_i = \alpha^*$ for $k < i \le N - m + 2$, and compute $\alpha_i$ by (5.3)–(5.4) for
            $i > \max(k, N - m + 2)$.
```

**5.2. Fast RILU for *m*-periodically monotone matrices.** We use the stopping criteria

$$\|\alpha^{(k)} - \alpha^{(k-1)}\| < \varepsilon \|\alpha^{(k)}\|$$

for the RILU recursions when $A$ is $m$-periodically monotone. The index $k$ here should not be larger than $(N - 2m + 2)/m$ since the $\alpha_i$'s are not periodic for $i > N - m + 2$. As soon as this criterion is satisfied, we set the remaining entries of the factor matrices in RILU by the periodical terms

$$\begin{aligned}
&\beta_{jm+i} = \beta_{km+i}, \quad &&\bar{\beta}_{jm+i} = \bar{\beta}_{km+i}, \quad &&j > k,\ jm + i \le n - 1,\\
&\gamma_{jm+i} = \gamma_{km+i}, \quad &&\bar{\gamma}_{jm+i} = \bar{\gamma}_{km+i}, \quad &&j > k,\ jm + i \le n - m + 1,\\
&\lambda_{jm+i} = \lambda_{km+i}, \quad &&\bar{\lambda}_{jm+i} = \bar{\lambda}_{km+i}, \quad &&j > k,\ jm + i \le n - m.
\end{aligned}$$

The substitution of the $\alpha_i$'s is restricted to the range $jm + i \le N - m + 2$; i.e., $\alpha_{jm+i} = \alpha_{km+i}$ for $jm + i \le N - m + 2$. The others are computed as in (5.3)–(5.4). If $mk > N - 2m - 2$, we complete the standard RILU recursions without the substitution.

```
ALGORITHM FRILU-II (FAST RILU FOR $m$-PERIODICALLY MONOTONE MATRICES).
Step 1. Partial RILU factorization.
        (1) Run the algorithm RILU until $\|\alpha^{(k)} - \alpha^{(k-1)}\| < \varepsilon\|\alpha^{(k)}\|$.
        (2) If $k > [(N-2m+2)/m]$, complete the RILU factorization and terminate.
Step 2. Periodically substituting.
            $\beta_{jm+i} = \beta_{km+i},\quad \bar{\beta}_{jm+i} = \bar{\beta}_{km+i},\quad j > k,\ jm + i \le n - 1,$
            $\lambda_{jm+i} = \lambda_{km+i},\quad \bar{\lambda}_{jm+i} = \bar{\lambda}_{km+i},\quad j > k,\ jm + i \le n - m,$
            $\gamma_{jm+i} = \gamma_{km+i},\quad \bar{\gamma}_{jm+i} = \bar{\gamma}_{km+i},\quad j > k,\ jm + i \le n - m + 1,$
            $\alpha_{jm+i} = \alpha_{km+i},\quad j > k,\ jm + i \le N - m + 2.$
Step 3. Compute $\alpha_i$ by (5.3)−(5.4) for $N - m + 3 \le i \le N$.
```

**6. Numerical results.** In this section, we give three numerical examples to illustrate the effectiveness of our fast algorithms for constructing RILU factorization. The examples are a constructed nine-diagonal matrix with diagonal constants, a real data coming from a bioengineering problem, and a nine-diagonal matrix from finite-difference PDE that is not diagonally monotone or periodically monotone. We check the uniform stability by the Newton iterations for the nonlinear system (3.8) with the initial guesses given in (3.9). The convergence criterion is (5.1)

$$\left\| \begin{pmatrix} t^{(k+1)} - t^{(k)} \\ \bar{t}^{(k+1)} - \bar{t}^{(k)} \end{pmatrix} \right\| < \tau,$$

where $(t^{(k)}, \bar{t}^{(k)})^T$ is the $k$th approximation computed by the Newton method. We use the stopping criteria

$$|\alpha_k - \alpha^*| < \varepsilon\alpha^*, \quad \|\alpha^{(k)} - \alpha^{(k-1)}\| < \varepsilon\|\alpha^{(k)}\|$$

with the accuracy parameter $\varepsilon$ for FRILU-I and FRILU-II, respectively.

[illegible] 1. The test matrix is a diagonally constant matrix

$$A = \{-\bar{e},\, -\bar{d},\, -\bar{c},\, -\bar{b},\, a,\, -b,\, -c,\, -d,\, -e\}$$

with

$$a = 13.9,\ b = 1,\ c = 2,\ d = 1,\ e = 1,\ \bar{e} = 2,\ \bar{d} = 2,\ \bar{c} = 3,\ \bar{b} = 2,$$

$m = 100$ and $N = 100,000$. The RILU factorization of $A$ is stable, sufficiently guaranteed by the uniform stability that can be checked by the Newton iterations. Starting with the initial guess shown in (3.9), the Newton method converges quickly for this matrix. For example, if $\omega = 0.5$, we obtain $t^* = 0.2268$, $\bar{t}^* = 0.1317$, and $\alpha^* = 11.687249$ approximately in the relative accuracy $\tau < 10^{-7}$ within 4 Newton iterations. The convergence slightly depends on the relaxation parameter $\omega$. In Table 2, we list the numbers of Newton iterations for different values of $\omega$ and the convergence accuracy $\tau$.

TABLE 2
*Numbers of Newton iterations for Example* 1.

| $\omega$ | log($\tau$) = -4 | -5 | -6 | -7 | -8 | -9 | -10 | -11 | -12 | -13 | -14 | -15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 0.2 | 3 | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 0.4 | 3 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 |
| 0.6 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 |
| 0.8 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 |
| 1.0 | 4 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 7 |

The recursive sequences of RILU converge fast for this example. In the right-hand panel of Figure 3, we plot the approximate errors of the first 2000 terms of the output sequence $\{\alpha_i\}$, $\{\beta_i\}$, $\{\gamma_i\}$, $\{\lambda_i\}$ of RILU with $\omega = 0$ to their limit values. The other output sequences have similar properties. It illustrates the consistent convergence behaviors in Theorem 2.4. In general, the convergence monotonically depends on the value of the relaxation $\omega$. See the left-hand panel of Figure 3 for the required iteration number $k$ of RILU($\omega$) with the relative accuracy $\epsilon = 10^{-12}$ for variant $\omega$.

This fast convergence implies that one can obtain the factorization at low computational cost. In Table 3, we list the number $k$ of the partial recursive steps of

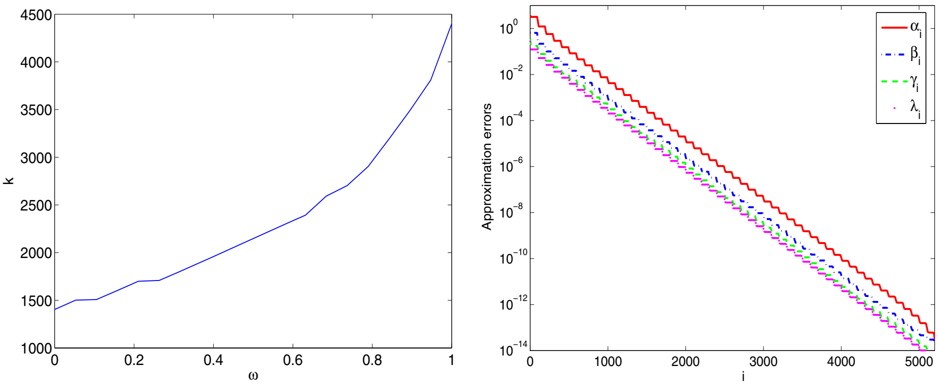


FIG. 3. *Left: convergence of RILU($\omega$) recursions. The required iteration number $k$ ($\epsilon = 10^{-12}$) monotonically depends on $\omega$. Right: consistent convergence of outputs. The approximation errors of the first* 5000 *terms of* $\{\alpha_i\}$, $\{\beta_i\}$, $\{\gamma_i\}$, $\{\lambda_i\}$ *to their limit values are plotted* ($\omega = 0$).

MILU (RILU with $\omega = 1$), flops, and CPU times (including the cost of the Newton iterations), and the total errors of FRILU-I for different values of $\varepsilon$, where the total error is defined by

$$Error = \|(\alpha, \beta, \gamma, \lambda, \bar{\beta}, \bar{\gamma}, \bar{\lambda}) - (\alpha^I, \beta^I, \gamma^I, \lambda^I, \bar{\beta}^I, \bar{\gamma}^I, \bar{\lambda}^I)\|,$$

where $\alpha$ and $\alpha^I$ are the sequences computed by the RILU and FRILU-I, respectively. Similar notations are used here for other outputs. The computational results show that the total error is proportional to the accuracy parameter $\varepsilon$. Notice that the storage cost is proportional to $k$. Compared with the costs of classical RILU listed in the right two columns of Table 3, FRILU is quite efficient for this example. It needs only about 1% of the cost of ILU in storage and flops, and 2% of CPU time. FRILU-I has similar convergence behavior for other values of $\omega$. Notice that the cost of FRILU is constant, while the CPU time of the standard RILU increases when $N$ is enlarged.

TABLE 3
*Computational costs of FRILU-I and MILU.*

| FRILU-I | | | | | MILU | |
|---|---|---|---|---|---|---|
| $\varepsilon$ | $k$ | Flops | CPU (s) | Error | Flops | CPU (s) |
| $10^{-3}$ | 901 | 41948 | 0.80 | $9.6 \times 10^{-2}$ | 43997013 | 10.98 |
| $10^{-4}$ | 1300 | 59903 | 0.80 | $9.0 \times 10^{-3}$ | | |
| $10^{-5}$ | 1699 | 77858 | 0.83 | $8.4 \times 10^{-4}$ | | |
| $10^{-6}$ | 2097 | 95768 | 0.84 | $8.0 \times 10^{-5}$ | | |
| $10^{-7}$ | 2493 | 113588 | 0.86 | $7.8 \times 10^{-6}$ | | |
| $10^{-8}$ | 2809 | 127808 | 0.88 | $1.2 \times 10^{-6}$ | | |
| $10^{-9}$ | 3206 | 145673 | 0.89 | $1.1 \times 10^{-7}$ | | |
| $10^{-10}$ | 3604 | 163583 | 0.95 | $1.1 \times 10^{-8}$ | | |

*Example* 2. The test matrix comes from a bioengineering problem.[1] It is a nine-point matrix of order $N = 9604$ and $m = 98$ with constant entries in the diagonal lines except for the periodic zeros $b_{km} = \bar{b}_{km} = 0$ $(1 \le k \le m-1)$, $c_{km+1} = \bar{c}_{km+1} = 0$ $(0 \le k \le m-1)$, and $e_{km} = \bar{e}_{km} = 0$ $(1 \le k \le m-2)$. So $A$ is a diagonally periodic

---

[1] This matrix, code name fv1, can be found at http://www.cise.ufl.edu/research/sparse/HBformat/Norris/.

matrix. Note that although $A$ is almost diagonally constant, all the output sequences are $m$-periodically monotone.

FRILU-II is also very efficient for this diagonally periodic matrix, due to the fast block-convergence of the output sequences. For example, within $k = 14$ block-iterations, $\{\alpha_i\}$ reaches the relative accuracy $\varepsilon = 10^{-10}$ when $\omega = 0.5$: $\|\alpha^{(k)} - \alpha^{(k-1)}\| < \varepsilon\|\alpha^{(k)}\|$, and the costs of FRILU-II in flops, CPU time, and storage are about 15% of those of RILU. In Table 4 we list the numbers $k$ of the block-iteration steps required to achieve the given relative accuracy $\varepsilon$ for different values of the relaxation parameter $\omega$. The number $k$ increases slightly as $\omega$ increases. Note that the number of componentwise recursive steps of FRILU-II is about $km$.

TABLE 4
*m-block iteration numbers of FRILU-II for Example* 2.

| $\omega$ | $\varepsilon = 10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ | $10^{-6}$ | $10^{-7}$ | $10^{-8}$ | $10^{-9}$ | $10^{-10}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0.0 | 3 | 4 | 5 | 6 | 8 | 9 | 10 | 11 | 12 |
| 0.2 | 3 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 13 |
| 0.4 | 4 | 5 | 6 | 7 | 9 | 10 | 11 | 12 | 14 |
| 0.6 | 4 | 5 | 6 | 8 | 9 | 11 | 12 | 13 | 15 |
| 0.8 | 4 | 5 | 7 | 8 | 10 | 11 | 13 | 14 | 16 |
| 1.0 | 4 | 6 | 7 | 9 | 11 | 12 | 14 | 16 | 17 |
| 1.2 | 4 | 6 | 8 | 10 | 11 | 13 | 15 | 17 | 19 |
| 1.4 | 4 | 6 | 8 | 10 | 12 | 15 | 17 | 19 | 21 |

Now we consider the efficiency of the incomplete LU factorization obtained by FRILU-II applied on the preconditioned GMRES algorithm for solving the linear system $Ax = f$. We randomly select a right-hand-side vector $f$ with components uniformly distributed in the interval $[-1, 1]$. The computational cost of FRILU-II can be ignored in this experiment. Its CPU time is only 0.06 seconds. The preconditioned GMRES algorithm starting at $x_0 = 0$ converges and achieves the accuracy $\|Ax - f\| < \epsilon$ with $\epsilon = 10^{-10}$ after 9 iterations and 0.25 seconds. We remark that the iteration number of GMRES with the preconditioner RILU($\omega$) is almost unchanged when the relaxation parameter $\omega$ varies in the interval $[0, 1]$. If the standard RILU is used in the preconditioned GMRES, the cost of the standard RILU is dominative in the entry computation. The CPU time of the standard RILU is 0.41 seconds. About half of the time is reduced by using FRILU-II, compared with the standard RILU preconditioning. We report that for this example the factorization ILUT$(p, \tau)$ [21, 22] costs much more than FILU-II or standard RILU, where $\tau$ is a dropping threshold of small components and $p$ the largest number of nonzero elements in each row of the factors $L$ or $U$. Table 5 summarizes the results for $p = 3, 5$ and $\tau = 0.01, 0.02, 0.03, 0.04, 0.05$. The column "ILUT sparsity" stands for the ratio $\frac{\mathrm{nnz}(L)+\mathrm{nnz}(U)-n}{\mathrm{nnz}(A)}$ of the total number of nonzero elements in both factors $L$ and $U$ and that in $A$. As $\tau$ increases, ILUT$(p, \tau)$ time decreases quickly. Meanwhile the preconditioning loses its efficiency gradually. Note that GMRES converges with 33 iterations without preconditioning.

[illegible] 3. This example shows the effectiveness of our fast algorithms for a more general matrix. Consider the convection-diffusion equation

$$-\Delta u + \sigma(xu_x + yu_y) - \rho u = f \quad \text{in} \quad \Omega = (0, 1) \times (0, 10)$$

with homogeneous Dirichlet boundary condition. Here $\sigma$ and $\rho$ are two positive constants. We discretize the PDE by a nine-point difference scheme with mesh size

TABLE 5

*ILUT(p, $\tau$) + GMRES with $\epsilon = 10^{-10}$ for fv1.*

| $\tau$ | ILUT sparsity | GMRES steps | CPU time ILUT | CPU time GMRES | CPU time Total |
|---|---|---|---|---|---|
| | | | $p = 3$ | | |
| 0.01 | 0.78 | 16 | 70.4 | 0.5 | 70.9 |
| 0.02 | 0.78 | 16 | 69.6 | 0.5 | 70.1 |
| 0.03 | 0.67 | 21 | 42.2 | 0.7 | 42.9 |
| 0.04 | 0.45 | 32 | 13.5 | 1.1 | 14.5 |
| 0.05 | 0.45 | 32 | 12.9 | 0.9 | 13.8 |
| | | | $p = 5$ | | |
| 0.01 | 1.22 | 9 | 71.6 | 0.3 | 71.9 |
| 0.02 | 1.00 | 10 | 72.6 | 0.3 | 72.9 |
| 0.03 | 0.78 | 20 | 42.5 | 0.5 | 43.0 |
| 0.04 | 0.56 | 32 | 13.8 | 0.8 | 14.7 |
| 0.05 | 0.56 | 32 | 13.6 | 0.9 | 14.5 |

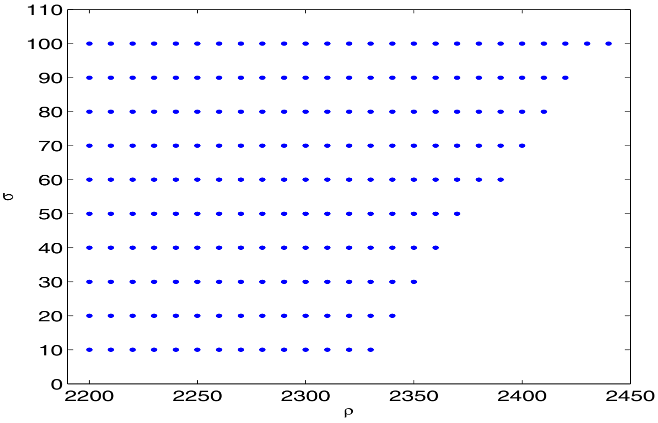


FIG. 4. *Stable region of the RILU factorization of $A(\rho,\sigma)$ in Example 3.*

$h = 1/(m+1)$. Let $\lceil x \rceil$ denote the smallest integer not less than $x$. The resulting nine-diagonal matrix $A = A(\rho,\sigma)$ of order $N = 10m^2$ has entries

$$a_i = 20 - 6\rho h^2, \quad d_i = 4 - 3\sigma h^2 \left\lceil \frac{i}{m} \right\rceil, \quad \bar{d}_i = 4 + 3\sigma h^2 \left\lceil \frac{i}{m} + 1 \right\rceil,$$
$$b_i = 4 - 3\sigma h^2 \mathrm{mod}(i,m), \quad \bar{b}_i = 4 + 3\sigma h^2(\mathrm{mod}(i,m) + 1),$$
$$c_i = 1, \quad \bar{c}_i = 1, \quad e_i = 1, \quad \bar{e}_i = 1,$$

except for the periodic zeros $b_{km} = \bar{b}_{km} = 0$ $(1 \le k \le m-1)$, $c_{km+1} = \bar{c}_{km+1} = 0$ $(0 \le k \le m-1)$, and $e_{km} = \bar{e}_{km} = 0$ $(1 \le k \le m-2)$. We set $m = 100$. The stability of the RILU factorization of $A(\rho,\sigma)$ depends on the parameters $\rho$ and $\sigma$. In Figure 4, we show a part of the stable region (marked by dots) in the $(\rho,\sigma)$-plane in which the matrix $A(\rho,\sigma)$ has stable RILU factorization with $\omega = 0.5$. All the $(\rho,\sigma)$ parameters used in this example are chosen in the stable region, and some of them are close to the boundary.

The matrix $A$ in this example is not diagonally periodically monotone. Ignoring the zeros appearing in the diagonals periodically, $\{d_i\}$ is piecewise monotone decreasing, while $\{\bar{d}_i\}$ is piecewise increasing for positive $\sigma$, though both $\{b_i\}$ and $\{\bar{b}_i\}$ are periodically constant. However, if $\rho$ is not large enough, the components change very slightly and the convergence of the output sequences is close to being periodical. We can apply algorithm FRILU-II on $A$ directly to compute an approximate RILU factorization. In Table 6 we show the pairs $(k,\, \epsilon_\alpha)$ for different parameters $(\rho, \sigma)$ chosen in the convergence region, where $k$ is the iteration number such that $\{\alpha^{(k)}\}$ achieves the given accuracy, and $\epsilon_\alpha = \|\alpha - \alpha^{II}\|_2/\|\alpha\|_2$ is the relative error of $\alpha^{II} = \alpha^{(k)}$ computed by FRILU-II with $\omega = 0.5$ and accuracy $\varepsilon = 10^{-4}$. $\{\beta_i^{II}\}$ has the same relative error as $\{\alpha_i^{II}\}$ in magnitude, while the accuracy of $\{\lambda_i^{II}\}$ and $\{\gamma_i^{II}\}$ are lower than the accuracy of $\{\alpha_i^{II}\}$ in general. We remark that the block approximation also depends on the parameters. If $(\rho, \sigma)$ tends to the boundary of the convergence region, then the approximate error $\|\alpha - \alpha^{II}\|_2$ increases, and the approximate block-convergence may no longer be maintained, though those parameters $\rho$ and $\sigma$ are chosen in the stable region. The loss of block-convergence within the given accuracy is indicated by "–" in Table 6.

TABLE 6
*Block-convergence* $(k,\, \epsilon_\alpha)$ *for Example* 3 *with* $\varepsilon = 10^{-4}$, $\omega = 0.5$.

| $\rho$ | $\sigma$=10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| 1500 | (13, 7.4e-4) | (12, 3.0e-3) | (12, 6.5e-3) | (12, 1.1e-2) | (11, 1.6e-2) |
| 1700 | (14, 8.4e-4) | (13, 3.4e-3) | (13, 7.5e-3) | (13, 1.3e-2) | (12, 1.8e-2) |
| 1900 | (16, 1.0e-3) | (15, 4.1e-3) | (14, 8.9e-3) | (14, 1.5e-2) | – |
| 2100 | (19, 1.3e-3) | (18, 5.4e-3) | (17, 1.1e-2) | – | – |
| 2200 | (23, 1.7e-3) | (21, 6.7e-3) | (19, 1.4e-2) | – | – |
| 2300 | (33, 2.7e-3) | (28, 9.8e-3) | – | – | – |
| 2320 | (41, 3.2e-3) | – | – | – | – |
| 2330 | (57, 3.3e-3) | – | – | – | – |

However, we can obtain an approximate RILU factorization with a higher accuracy by applying FRILU-II on a true diagonally periodic matrix $A_c$. Such a matrix $A_c$ can be obtained from $A$ by replacing $\{d_i\}$ and $\{\bar{d}_i\}$ by the constant $d_1 = 4 - 3\sigma h^2$ and $\bar{d}_1 = 4 + 6\sigma h^2$, respectively. Let $\{\alpha_i^c\}$ and the other outputs with subscript $c$ be the computed sequences by FRILU-II for $A_c$. Since there are input errors $d_i - d_1$ and $\bar{d}_i - \bar{d}_1$ in $\{d_i^c\}$ and $\{\bar{d}_i{}^c\}$, $\{\lambda_i^c\}$ and $\{\gamma_i^c\}$ have relatively large errors to $\{\lambda_i\}$ and $\{\gamma_i\}$. However $\{\alpha_i^c\}$, $\{\beta_i^c\}$, and $\{\bar{\beta}_i^c\}$ still have acceptable accuracy if $\sigma$ is not large. In that case, we can improve $\lambda_i^c$ and $\gamma_i^c$ by adding back the input errors to the outputs $\lambda_i^c$ and $\gamma_i^c$ with the refinement

$$\lambda_i^R = \lambda_i^c + (d_i - d_1), \quad \bar{\lambda}_i^R = \bar{\lambda}_i^c + (\bar{d}_i - \bar{d}_1),$$

$$\gamma_i^R = \gamma_i^c + (d_{i-1} - d_1)\frac{\bar{\beta}_{i-1}^c}{\alpha_{i-1}^c}, \quad \bar{\gamma}_i^R = \bar{\gamma}_i^c + (\bar{d}_{i-1} - \bar{d}_1)\frac{\beta_{i-1}^c}{\alpha_{i-1}^c}.$$

This refinement gives acceptable accuracy of the approximate RILU factorization. Table 7 shows the accuracy of the computed $\{\lambda_i\}$ and $\{\gamma_i\}$ by applying the algorithm FRILU-II to $A$ or the approximation $A_c$, and the refinement discussed above, marked by I, II, and III in the second column of Table 7, respectively. Note that for some values of $\rho$ and $\sigma$, FRILU-II fails if it is applied on $A$ directly.

TABLE 7
*Relative accuracy of* $\{\lambda_i\}$ *and* $\{\gamma_i\}$ *computed by* (I) *FRILU-II for* $A$, (II) *FRILU-II for* $A_c$, *and* (III) *the refinement method* ($\tau = 10^{-4}$, $\omega = 0.5$).

| | | $\{\lambda_i\}$ | | | | | $\{\gamma_i\}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\rho$ | | $\sigma$=10 | 20 | 30 | 40 | 50 | $\sigma$=10 | 20 | 30 | 40 | 50 |
| 1500 | I | 3.3e-2 | 6.9e-2 | 1.1e-1 | 1.5e-1 | 2.0e-1 | 2.0e-2 | 4.4e-2 | 7.1e-2 | 9.9e-2 | 1.3e-1 |
| 1700 | | 3.2e-2 | 6.8e-2 | 1.1e-1 | 1.5e-1 | 1.9e-1 | 2.0e-2 | 4.5e-2 | 7.1e-2 | 1.0e-1 | 1.3e-1 |
| 1900 | | 3.1e-2 | 6.5e-2 | 1.0e-1 | 1.4e-1 | – | 2.0e-2 | 4.4e-2 | 7.2e-2 | 1.0e-1 | – |
| 2100 | | 2.9e-2 | 6.2e-2 | 9.8e-2 | – | – | 2.0e-2 | 4.4e-2 | 7.2e-2 | – | – |
| 2200 | | 2.7e-2 | 5.9e-2 | 9.5e-2 | – | – | 1.9e-2 | 4.4e-2 | 7.3e-2 | – | – |
| 2300 | | 2.2e-2 | 5.1e-2 | – | – | – | 1.7e-2 | 4.3e-2 | – | – | – |
| 2320 | | 1.8e-2 | – | – | – | – | 1.5e-2 | – | – | – | – |
| 2330 | | 1.2e-2 | – | – | – | – | 1.1e-2 | – | – | – | – |
| 1500 | II | 4.0e-2 | 8.2e-2 | 1.3e-1 | 1.8e-1 | 2.3e-1 | 2.5e-2 | 5.3e-2 | 8.4e-2 | 1.2e-1 | 1.5e-1 |
| 1700 | | 4.0e-2 | 8.2e-2 | 1.3e-1 | 1.8e-1 | 2.3e-1 | 2.5e-2 | 5.4e-2 | 8.6e-2 | 1.2e-1 | 1.6e-1 |
| 1900 | | 4.0e-2 | 8.2e-2 | 1.3e-1 | 1.8e-1 | 2.3e-1 | 2.6e-2 | 5.5e-2 | 8.8e-2 | 1.2e-1 | 1.6e-1 |
| 2100 | | 4.0e-2 | 8.2e-2 | 1.3e-1 | 1.8e-1 | 2.3e-1 | 2.6e-2 | 5.8e-2 | 9.3e-2 | 1.3e-1 | 1.7e-1 |
| 2200 | | 4.0e-2 | 8.2e-2 | 1.3e-1 | 1.8e-1 | 2.3e-1 | 2.7e-2 | 6.0e-2 | 9.7e-2 | 1.4e-1 | 1.8e-1 |
| 2300 | | 4.0e-2 | 8.3e-2 | 1.3e-1 | 1.8e-1 | 2.3e-1 | 3.0e-2 | 6.7e-2 | 1.1e-1 | 1.5e-1 | 1.9e-1 |
| 2320 | | 4.0e-2 | 8.4e-2 | 6.6e-1 | 1.9e-1 | 2.5e-1 | 3.4e-2 | 7.6e-2 | 4.5e+0 | 3.6e-1 | 7.7e-1 |
| 2330 | | 2.6e-1 | 1.3e-1 | 1.4e-1 | 2.5e-1 | 5.7e-1 | 2.1e+0 | 7.9e-1 | 4.8e-1 | 1.2e+0 | 3.7e+0 |
| 1500 | III | 1.1e-4 | 4.6e-4 | 1.1e-3 | 1.9e-3 | 2.9e-3 | 8.0e-4 | 3.2e-3 | 6.8e-3 | 1.1e-2 | 1.6e-2 |
| 1700 | | 1.2e-4 | 5.4e-4 | 1.2e-3 | 2.2e-3 | 3.4e-3 | 9.3e-4 | 3.8e-3 | 8.0e-3 | 1.3e-2 | 1.9e-2 |
| 1900 | | 1.6e-4 | 6.8e-4 | 1.5e-3 | 2.7e-3 | 4.0e-3 | 1.2e-3 | 4.6e-3 | 9.7e-3 | 1.6e-2 | 2.2e-2 |
| 2100 | | 2.3e-4 | 9.4e-4 | 2.1e-3 | 3.5e-3 | 5.1e-3 | 1.6e-3 | 6.3e-3 | 1.3e-2 | 2.0e-2 | 2.7e-2 |
| 2200 | | 3.0e-4 | 1.2e-3 | 2.6e-3 | 4.2e-3 | 6.0e-3 | 2.2e-3 | 8.2e-3 | 1.6e-2 | 2.4e-2 | 3.2e-2 |
| 2300 | | 6.1e-4 | 2.1e-3 | 4.0e-3 | 5.9e-3 | 8.0e-3 | 4.3e-3 | 1.4e-2 | 2.5e-2 | 3.4e-2 | 4.3e-2 |
| 2320 | | 1.1e-3 | 3.5e-3 | 6.5e-1 | 4.5e-2 | 1.0e-1 | 8.1e-3 | 2.3e-2 | 3.8e+0 | 2.7e-1 | 5.7e-1 |
| 2330 | | 2.5e-1 | 1.0e-1 | 6.2e-2 | 1.8e-1 | 5.3e-1 | 2.0e+0 | 7.3e-1 | 4.1e-1 | 9.9e-1 | 2.7e+0 |

**Acknowledgments.** We want to thank the anonymous referees for their careful reading of this paper. Their insightful suggestions and comments have greatly improved the presentation of this paper.

## REFERENCES

[1] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problem*, Academic Press, Orlando, FL, 1984.

[2] O. Axelsson and G. Lindskog, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math., 48 (1986), pp. 479–498.

[3] M. Benzi, J. C. Haws, and M. Tůma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.

[4] M. Bhuruth, M. K. Jain, and A. Gopaul, *Preconditioned iterative methods for the nine-point approximation to the convection-diffusion equation*, J. Comput. Appl. Math., 138 (2002), pp. 73–92.

[5] A. M. Bruaset, A. Tveito, and R. Winther, *On the stability of relaxed incomplete LU factorizations*, Math. Comp., 54 (1990), pp. 701–719.

[6] E. Chow and Y. Saad, *Experimental study of ILU preconditioners for indefinite matrices*, J. Comput. Appl. Math., 86 (1997), pp. 387–414.

[7] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.

[8] T. Dupont, R. P. Kendall, and H. H. Rachford, *An approximate factorization procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal., 5 (1968), pp. 559–573.

[9] H. C. Elman, *A stability analysis of incomplete LU factorizations*, Math. Comp., 47 (1986), pp. 191–217.

[10] H. C. Elman, *Relaxed and stabilized incomplete factorizations for nonself-adjoint linear systems*, BIT, 29 (1989), pp. 890–915.

[11] G.-D. Gu, *Some conditions for existence and stability of relaxed incomplete LU factorizations*, Appl. Numer. Math., 38 (2001), pp. 105–121.

[12] I. Gustafsson, *A class of first order factorization methods*, BIT, 18 (1978), pp. 142–156.

[13] S. Karaa and J. Zhang, *A note on convergence of line iterative methods for a nine-point matrix*, Appl. Math. Lett., 15 (2002), pp. 495–503.

[14] C. F. Karney, *Fokker-Planck and quasilinear codes*, Comput. Phys. Rep., 4 (1986), pp. 183–244.

[15] T. Manteuffel, *An incomplete factorization technique for positive definite linear systems*, Math. Comp., 34 (1980), pp. 473–497.

[16] J. A. Meijerink, and H. A. van der Vorst, *An interactive solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[17] J. A. Meijerink and H. A. van der Vorst, *Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems*, J. Comput. Phys., 44 (1981), pp. 134–155.

[18] N. Munksgaard, *Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients*, ACM Trans. Math. Software, 6 (1980), pp. 206–219.

[19] M. Olschowka and A. Neumaier, *A new pivoting strategy for Gaussian elimination*, Linear Algebra Appl., 240 (1996), pp. 131–151.

[20] Y. Peysson and M. Shoucri, *An approximate factorization procedure for solving nine-point elliptic difference equations. Application for a fast 2-D relativistic Fokker-Planck solver*, Comput. Phys. Comm., 109 (1998), pp. 55–80.

[21] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, Boston, 1996.

[22] Y. Saad, *ILUT: A dual threshold incomplete LU preconditioner*, Numer. Linear Algebra Appl., 1 (1994), pp. 387–402.

[23] V. Simoncini and D. B. Szyld, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2002), pp. 2219–2239.

[24] R. S. Varga, E. B. Saff, and V. Mehrman, *Incomplete factorizations of matrices and connections with H-matrices*, SIAM J. Numer. Anal., 17 (1980), pp. 787–793.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 554–565

# SIGN PATTERNS THAT ALLOW A POSITIVE OR NONNEGATIVE LEFT INVERSE*

IN-JAE KIM†, D. D. OLESKY‡, B. L. SHADER§, AND P. VAN DEN DRIESSCHE¶

**Abstract.** An $m$ by $n$ sign pattern $\mathcal{S}$ is an $m$ by $n$ matrix with entries in $\{+, -, 0\}$. Such a sign pattern allows a positive (resp., nonnegative) left inverse, provided that there exist an $m$ by $n$ matrix $A$ with the sign pattern $\mathcal{S}$ and an $n$ by $m$ matrix $B$ with only positive (resp., nonnegative) entries satisfying $BA = I_n$, where $I_n$ is the $n$ by $n$ identity matrix. For $m > n \geq 2$, a characterization of $m$ by $n$ sign patterns with no rows of zeros that allow a positive left inverse is given. This leads to a characterization of all $m$ by $n$ sign patterns with $m \geq n \geq 2$ that allow a positive left inverse, giving a generalization of the known result for the square case, which involves a related bipartite digraph. For $m \geq n$, $m$ by $n$ sign patterns with all entries in $\{+, 0\}$ and $m$ by 2 sign patterns with $m \geq 2$ that allow a nonnegative left inverse are characterized, and some necessary or sufficient conditions for a general $m$ by $n$ sign pattern to allow a nonnegative left inverse are presented.

**Key words.** bipartite digraph, nonnegative left inverse, positive left inverse, positive left nullvector, sign pattern, strong Hall

**AMS subject classifications.** 15A09, 15A48, 05C20, 05C50

**DOI.** 10.1137/060660916

## 1. Introduction.

An $m$ by $n$ *sign pattern* $\mathcal{S} = [s_{ij}]$ is an $m$ by $n$ matrix with entries in $\{+, -, 0\}$. If a sign pattern $\mathcal{S}$ has all entries in $\{+, 0\}$, then $\mathcal{S}$ is a *nonnegative* sign pattern. A *subpattern* of $\mathcal{S}$ is an $m$ by $n$ sign pattern $\mathcal{U} = [u_{ij}]$ such that $u_{ij} = 0$ whenever $s_{ij} = 0$. If $\mathcal{U}$ is a subpattern of $\mathcal{S}$, then $\mathcal{S}$ is a *superpattern* of $\mathcal{U}$. The *sign pattern class* $Q(\mathcal{S})$ of an $m$ by $n$ sign pattern $\mathcal{S}$ is the set of $m$ by $n$ matrices $A = [a_{ij}]$ such that $\text{sgn}(a_{ij}) = s_{ij}$ for all $i, j$. If $A \in Q(\mathcal{S})$, then $A$ is a *realization* of $\mathcal{S}$.

Let $A = [a_{ij}]$ be an $m$ by $n$ matrix. If each entry of $A$ is positive (resp., nonnegative), then $A$ is *positive* (resp., *nonnegative*), written $A > 0$ (resp., $A \geq 0$). A *left inverse* of an $m$ by $n$ matrix $A$ is an $n$ by $m$ matrix $B$ such that $BA = I_n$, where $I_n$ denotes the $n$ by $n$ identity matrix. If $B > 0$, then $B$ is a *positive* left inverse (abbreviated as *PLI*) of $A$. If $B \geq 0$, then $B$ is a *nonnegative* left inverse (abbreviated as *NLI*) of $A$. In general, neither a PLI nor an NLI of $A$ is unique. It is easily verified that $A$ has a left inverse if and only if $\operatorname{rank} A = n$; thus, if $A$ has a left inverse, then necessarily $m \geq n$. An $m$ by $n$ sign pattern $\mathcal{S}$ *allows a positive* (resp., *nonnegative*) *left inverse*, provided there exist $A \in Q(\mathcal{S})$ and a matrix $B > 0$ (resp., $B \geq 0$) such that $BA = I_n$. Note that if $P_1$ and $P_2$ are permutation matrices, then $\mathcal{S}$ allows a PLI (resp., an NLI) if and only if $P_1 \mathcal{S} P_2$ allows a PLI (resp., an NLI).

---

[*] Received by the editors May 25, 2006; accepted for publication (in revised form) by R. Bhatia October 5, 2006; published electronically April 20, 2007.
http://www.siam.org/journals/simax/29-2/66091.html

[†] Department of Mathematics and Statistics, Minnesota State University, Mankato, MN 56001 (in-jae.kim@mnsu.edu). The research of this author was done while he was a postdoctoral fellow at the University of Victoria.

[‡] Department of Computer Science, University of Victoria, P.O. Box 3055, Victoria, BC, Canada V8W 3P6 (dolesky@cs.uvic.ca). The research of this author was supported in part by an NSERC Discovery Grant.

[§] Department of Mathematics, University of Wyoming, Laramie, WY 82071 (bshader@uwyo.edu).

[¶] Department of Mathematics and Statistics, University of Victoria, P.O. Box 3045, Victoria, BC, Canada V8W 3P4 (pvdd@math.uvic.ca). The research of this author was supported in part by an NSERC Discovery Grant.

A motivation for studying PLIs and NLIs comes from determining the qualitative behavior of solutions of $A^T x = b$ with $A$ an $m$ by $n$ matrix; see, for example, [2, Chapter 1] and [5] for applications in economics. Specifically, $A$ has a PLI (resp., an NLI) if and only if for each $n$ by 1 nonzero vector $b \geq 0$ there exists an $m$ by 1 vector $x > 0$ (resp., $x \geq 0$) satisfying $A^T x = b$ or equivalently $x^T A = b^T$; see Proposition 4.1 for a proof.

Square sign patterns with entries in $\{+, -\}$ that allow a positive (left) inverse are characterized in [6], and this characterization is extended to arbitrary square sign patterns in [4]. These results are summarized in [2, section 9.2]. In section 2, we characterize nonsquare sign patterns that allow a PLI, and combine the square and nonsquare characterizations. In section 3, we discuss sign patterns that allow an NLI. More specifically, we characterize nonnegative sign patterns and $m$ by 2 sign patterns with $m \geq 2$ that allow an NLI, and present some necessary or sufficient conditions for general $m$ by $n$ sign patterns with $m \geq n$ to allow an NLI. We conclude with some remarks in section 4.

**2. Positive left inverses.** We begin this section with a necessary and sufficient condition for a column sign pattern to allow a PLI or an NLI.

PROPOSITION 2.1. [illegible] $\mathcal{S} = (s_1, s_2, \ldots, s_m)^T$ [illegible] $m$ [illegible] 1 [illegible]

(i) $\mathcal{S}$ [illegible] $+$ [illegible]
(ii) $\mathcal{S}$ [illegible]
(iii) $\mathcal{S}$ [illegible]

[illegible]. Suppose there is an index $i \in \{1, 2, \ldots, m\}$ with $s_i = +$. For $j \in \{1, \ldots, m\}$, set

$$a_j = \begin{cases} 1 & \text{if } j \neq i \text{ and } s_j = +, \\ -1 & \text{if } j \neq i \text{ and } s_j = -, \\ 0 & \text{if } j \neq i \text{ and } s_j = 0, \\ 1 + \sum_{k \neq i} |a_k| & \text{if } j = i. \end{cases}$$

Then $A = (a_1, \ldots, a_m)^T \in Q(\mathcal{S})$, and $(1, 1, \ldots, 1)A = 1 + \sum_{k \neq i} (|a_k| + a_k) = c > 0$. This implies that $\frac{1}{c}(1, 1, \ldots, 1)$ is a PLI of $A$. Thus, $\mathcal{S}$ allows a PLI.

It is clear that (ii) implies (iii). Next, suppose that the sign pattern $\mathcal{S}$ allows an NLI. Then there exist $A = (a_1, \ldots, a_m)^T \in Q(\mathcal{S})$ and $B = (b_1, \ldots, b_m) \geq 0$ such that $BA = 1$, i.e., $\sum_{j=1}^m b_j a_j = 1 > 0$. This implies that there exists an $i$ with $b_i a_i > 0$. Since $b_i \geq 0$, it follows that $b_i > 0$; hence $a_i > 0$ and thus $s_i = +$. $\square$

We now consider $m \geq n \geq 2$. The following two lemmas give necessary conditions for a sign pattern to allow a PLI.

LEMMA 2.2. [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible] $+$ [illegible] $-$ [illegible]

[illegible]. Suppose that there exist $A \in Q(\mathcal{S})$ and an $n$ by $m$ positive matrix $B$ such that $BA = I_n$. Let $i \in \{1, 2, \ldots, n\}$. Since the $(i, i)$-entry of $BA$ is 1 and each entry of $B$ is positive, it follows that some entry in column $i$ of $A$ is positive. Hence, column $i$ of $\mathcal{S}$ has a $+$ entry.

Since $n \geq 2$, there exists $j \in \{1, \ldots, n\}$ with $j \neq i$. The $(j, i)$-entry of $BA$ is 0, so since $B > 0$ and at least one entry of column $i$ of $A$ is positive, it follows that at least one entry of column $i$ of $A$ must be negative. Thus, column $i$ of $\mathcal{S}$ has a $-$ entry. $\square$

An $m$ by $n$ sign pattern $\mathcal{S}$ with $n \geq 2$ is [illegible], provided that for every nonempty proper subset $\gamma$ of $\{1, 2, \ldots, n\}$ the submatrix of $\mathcal{S}$ consisting of the columns indexed by $\gamma$ has nonzero entries in at least $|\gamma|+1$ rows (see [3]). Note that if $\mathcal{S}$ is strong Hall, then necessarily $m \geq n$. Also, for $m \geq n$, $\mathcal{S}$ is not strong Hall if and only if there exist permutation matrices $P_1$ and $P_2$ such that

$$P_1 \mathcal{S} P_2 = \begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} \\ O & \mathcal{S}_{22} \end{bmatrix}, \tag{2.1}$$

where $\mathcal{S}_{11}$ is a $k$ by $\ell$ sign pattern for some integers $k$, $\ell$ with $n > \ell \geq 1$ and $k \leq \ell$.

LEMMA 2.3. [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible]

[illegible]. To prove the contrapositive, assume that $\mathcal{S}$ is not strong Hall. If $m < n$, then it is clear that $\mathcal{S}$ does not allow a PLI. Otherwise, without loss of generality, we may assume that $\mathcal{S}$ has the form (2.1). If $k < \ell$, then the first $\ell$ columns of each realization of $\mathcal{S}$ are linearly dependent, and hence $\mathcal{S}$ does not allow a PLI.

Otherwise, $k = \ell < n$. Suppose that there exists a matrix $A = \left[\begin{smallmatrix} A_{11} & A_{12} \\ O & A_{22} \end{smallmatrix}\right] \in Q(\mathcal{S})$ with a left inverse $B = \left[\begin{smallmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{smallmatrix}\right]$, where $B_{11}$ is an $\ell$ by $\ell$ matrix. Clearly, the $\ell$ by $\ell$ matrix $A_{11}$ is invertible, and by $BA = I_n$, it follows that $\left[\begin{smallmatrix} B_{21} & B_{22} \end{smallmatrix}\right] \left[\begin{smallmatrix} A_{11} \\ O \end{smallmatrix}\right] = O$. Thus, $B_{21}A_{11} = O$, and since $A_{11}$ is invertible, the $(n-\ell)$ by $\ell$ matrix $B_{21} = O$. Since $n - \ell \geq 1$ and $\ell \geq 1$, every left inverse of a matrix in $Q(\mathcal{S})$ has a zero entry, and hence $\mathcal{S}$ does not allow a PLI. $\square$

Note that if $\mathcal{S}$ is a square sign pattern of order $n \geq 2$, then $\mathcal{S}$ is strong Hall if and only if $\mathcal{S}$ is fully indecomposable (see [3]), and $\mathcal{S}$ allows a PLI if and only if $\mathcal{S}$ allows a positive inverse. The next theorem, first proved in [4], provides a characterization of square sign patterns that allow a positive inverse. In order to recall this characterization, we use the following definition as in [1] and [2]. Let $\mathcal{S} = [s_{ij}]$ be an $m$ by $n$ sign pattern. The [illegible] $D(\mathcal{S})$ of $\mathcal{S}$ is the digraph with row vertices $u_1, \ldots, u_m$, column vertices $v_1, \ldots, v_n$, an arc $u_i \to v_j$ if $s_{ij} = +$, and an arc $v_j \to u_i$ if $s_{ij} = -$. Note that there exists at most one arc between $u_i$ and $v_j$.

THEOREM 2.4 (see [2, Theorem 9.2.1]). [illegible] $n$ [illegible] $n$ [illegible] $\mathcal{S}$ [illegible] $n \geq 2$ [illegible] ( [illegible] ) [illegible] $\mathcal{S}$ [illegible] $D(\mathcal{S})$ [illegible] $\mathcal{S}$ [illegible]

Let $\mathcal{S}$ be an $m$ by $n$ sign pattern and let $D(\mathcal{S})$ be its bipartite digraph. A [illegible] of $D(\mathcal{S})$ is a maximal strongly connected subdigraph of $D(\mathcal{S})$. If $\alpha$ is a strong component of $D(\mathcal{S})$, then $|\alpha|$ denotes the number of vertices in $\alpha$.

[illegible] 2.5. Let $\alpha$ be a strong component of $D(\mathcal{S})$. Since $D(\mathcal{S})$ is a bipartite digraph with no cycles of length 2, it follows that if $|\alpha| \geq 2$, then $\alpha$ has at least two row vertices and at least two column vertices.

Let $\alpha_1, \alpha_2, \ldots, \alpha_t$ be the strong components of $D(\mathcal{S})$. The [illegible] digraph $CD(\mathcal{S})$ of $\mathcal{S}$ has vertices $\alpha_1, \alpha_2, \ldots, \alpha_t$ and an arc $\alpha_i \to \alpha_j$ if and only if $i \neq j$ and $D(\mathcal{S})$ has at least one arc from a vertex in $\alpha_i$ to a vertex in $\alpha_j$. A strong component $\alpha_i$ of $D(\mathcal{S})$ is a [illegible] if there is no arc coming into $\alpha_i$ in $CD(\mathcal{S})$ and there is at least one arc coming out of $\alpha_i$ in $CD(\mathcal{S})$; $\alpha_i$ is a [illegible] if there is no arc coming out of $\alpha_i$ in $CD(\mathcal{S})$ and there is at least one arc coming into $\alpha_i$ in $CD(\mathcal{S})$; and $\alpha_i$ is [illegible] if there are no arcs coming into or out of $\alpha_i$ in $CD(\mathcal{S})$.

LEMMA 2.6. [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $+$ [illegible] $-$ [illegible] $D(\mathcal{S})$:

(i) [illegible] $D(\mathcal{S})$ [illegible]

(ii) [illegible]

[illegible]. (i) Let $\alpha$ be a sink or source strong component. If $|\alpha| = 1$, then since each column of $\mathcal{S}$ has a $+$ and a $-$ entry, it follows that no sink or source strong component consists of exactly one column vertex. Hence, $\alpha$ is a row vertex. If $|\alpha| \geq 2$, then Remark 2.5 implies that $\alpha$ has at least one row vertex.

(ii) By the assumptions on the rows and columns of $\mathcal{S}$, there is no isolated strong component with exactly one vertex. Hence, by Remark 2.5, each isolated strong component has at least two row vertices. $\square$

Let $A$ be an $m$ by $n$ matrix with $m \geq n$. If there exists an $m$ by 1 vector $y > 0$ satisfying $y^T A = 0$, then $y^T$ is a [illegible] nullvector of $A$. The following theorem gives a characterization of nonsquare sign patterns with no rows of zeros that allow a PLI. Note that conditions for such a sign pattern to allow a PLI are weaker than those for square sign patterns (Theorem 2.4), although the bipartite digraph is used in our proof for a nonsquare sign pattern.

THEOREM 2.7. [illegible] $m > n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible]

(i) [illegible] $A \in Q(\mathcal{S})$ [illegible]

(ii) $\mathcal{S}$ [illegible]

(iii) [illegible] $\mathcal{S}$ [illegible] $+$ [illegible] $-$ [illegible] $\mathcal{S}$ [illegible]

[illegible]. Clearly, (i) implies (ii). By Lemmas 2.2 and 2.3, (ii) implies (iii).

To prove that (iii) implies (i), assume that $\mathcal{S}$ is strong Hall and that $\mathcal{S}$ has a $+$ and a $-$ entry in each column. We claim that it suffices to show that there exists an $m$ by $(m-n)$ sign pattern $\mathcal{C}$ so that the $m$ by $m$ sign pattern $[\mathcal{S} \mid \mathcal{C}]$ allows a positive (left) inverse. To prove this claim, suppose there exists an $m$ by $m$ matrix $[A \mid C] \in Q([\mathcal{S} \mid \mathcal{C}])$ with a positive (left) inverse $\left[\begin{smallmatrix} B_1 \\ B_2 \end{smallmatrix}\right]$ where $B_1$ is an $n$ by $m$ positive matrix and $B_2$ is an $(m-n)$ by $m$ positive matrix. Then $B_1 A = I_n$ and hence $B_1$ is a PLI of $A$, implying that $\mathcal{S}$ allows a PLI. In addition, since $B_2 A = O$ and $B_2$ has at least one positive row, $A$ has a positive left nullvector. Therefore, by Theorem 2.4, it suffices to find an $m$ by $(m-n)$ sign pattern $\mathcal{C}$ such that the $m$ by $m$ sign pattern $[\mathcal{S} \mid \mathcal{C}]$ is strong Hall and its bipartite digraph $D([\mathcal{S} \mid \mathcal{C}])$ is strongly connected.

Consider the bipartite digraph $D(\mathcal{S})$ of $\mathcal{S}$. Let $\alpha_1, \alpha_2, \ldots, \alpha_t$ be its strong components, where $\alpha_1, \ldots, \alpha_k$ are sinks, $\alpha_{k+1}, \ldots, \alpha_{k+\ell}$ are sources, $\alpha_{k+\ell+1}, \ldots, \alpha_{k+\ell+r}$ are isolated, and $\alpha_{k+\ell+r+1}, \ldots, \alpha_t$ are neither sinks, sources, nor isolated strong components. By Lemma 2.6 (i), each sink and source strong component has a row vertex. Let $r_i$ be a fixed row vertex of $\alpha_i$ for each $i \in \{1, \ldots, k+\ell\}$. Also, by Lemma 2.6 (ii), each isolated strong component has at least two row vertices. Let $r_i^+, r_i^-$ be distinct fixed row vertices of $\alpha_i$ for each $i \in \{k+\ell+1, \ldots, k+\ell+r\}$. Let $\mathcal{C}_{n+1}$ be the $m$ by 1 column sign pattern with nonzero $j$th coordinate:

$$\begin{cases} + & \text{if } u_j \in \{r_1, \ldots, r_k\} \cup \{r_{k+\ell+1}^-, \ldots, r_{k+\ell+r}^-\}, \\ - & \text{if } u_j \in \{r_{k+1}, \ldots, r_{k+\ell}\} \cup \{r_{k+\ell+1}^+, \ldots, r_{k+\ell+r}^+\}, \\ + & \text{otherwise.} \end{cases} \tag{2.2}$$

Then $D([\mathcal{S} \mid \mathcal{C}_{n+1}])$ is obtained from $D(\mathcal{S})$ by appending a new column vertex $c_{n+1}$, and arcs $r_j \to c_{n+1}$ if $r_j$ is in a sink component; $c_{n+1} \to r_j$ if $r_j$ is in a source component; $r_j^- \to c_{n+1}$ and $c_{n+1} \to r_j^+$ if $r_j^-$ and $r_j^+$ are in the same isolated component; as well as some additional arcs coming into vertex $c_{n+1}$.

To prove that $D([\mathcal{S} \mid \mathcal{C}_{n+1}])$ is strongly connected, we show that for each vertex $w$ of $D(\mathcal{S})$ there exists in $D([\mathcal{S} \mid \mathcal{C}_{n+1}])$ a walk from $c_{n+1}$ to $w$ and a walk from $w$ to $c_{n+1}$. Note that if $w$ is not in an isolated strong component of $D(\mathcal{S})$, then there is a walk from $w$ to a vertex in a sink strong component $\alpha_i$ of $D(\mathcal{S})$ $(i \in \{1, \ldots, k\})$. Since $\alpha_i$ is strongly connected, this walk from $w$ can be extended to the fixed row vertex $r_i$ of $\alpha_i$. By (2.2), there is an arc $r_i \to c_{n+1}$ in $D([\mathcal{S} \mid \mathcal{C}_{n+1}])$. Hence, there is a walk from $w$ to $c_{n+1}$. Similarly, there is a walk from $c_{n+1}$ to $w$.

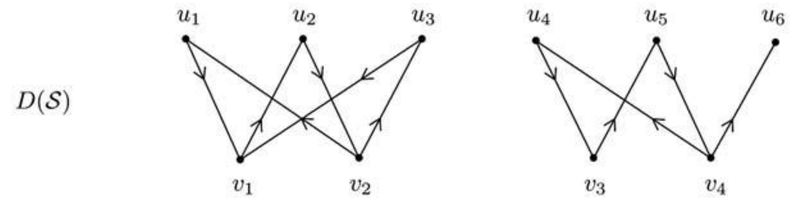
Next, suppose that $w$ is a vertex in an isolated strong component $\alpha_i$ in $D(\mathcal{S})$ ($i \in \{k+\ell+1,\ldots,k+\ell+r\}$). Since $\alpha_i$ is strongly connected, there is a walk from $w$ to the fixed row vertex $r_i^-$ of $\alpha_i$. By (2.2), there are arcs $r_i^- \to c_{n+1}$ and $c_{n+1} \to r_i^+$ in $D([\mathcal{S} \mid \mathcal{C}_{n+1}])$. Since $\alpha_i$ is strongly connected, there is a walk from $r_i^+$ to $w$. Thus, there exist a walk from $w$ to $c_{n+1}$ and a walk from $c_{n+1}$ to $w$.

Finally, define $\mathcal{C}_{n+2},\ldots,\mathcal{C}_m$ to be $m$ by 1 column sign patterns, each having no zeros, at least one $+$, and at least one $-$ entry. Then it is easily verified that $D([\mathcal{S} \mid \mathcal{C}_{n+1} \mid \ldots \mid \mathcal{C}_m])$ is strongly connected. Since $\mathcal{S}$ is strong Hall and $[\mathcal{C}_{n+1} \mid \ldots \mid \mathcal{C}_m]$ has no zeros, it is clear that $[\mathcal{S} \mid \mathcal{C}_{n+1} \mid \ldots \mid \mathcal{C}_m]$ is strong Hall, completing the proof. $\square$

Example 2.8. Consider the 6 by 4 sign pattern

$$\mathcal{S} = \begin{bmatrix} + & - & 0 & 0 \\ - & + & 0 & 0 \\ + & - & 0 & 0 \\ 0 & 0 & + & - \\ 0 & 0 & - & + \\ 0 & 0 & 0 & - \end{bmatrix}$$

with



Each column of $\mathcal{S}$ has a $+$ and a $-$ entry, and $\mathcal{S}$ is strong Hall. Thus, by Theorem 2.7, $\mathcal{S}$ allows a PLI. However, $D(\mathcal{S})$ is not strongly connected, illustrating a distinction between the nonsquare and square cases (see Theorem 2.4). In fact, $D(\mathcal{S})$ has one sink strong component $\alpha_1$ that consists of vertex $u_6$, one source strong component $\alpha_2$ with vertices $u_4, u_5, v_3, v_4$, and one isolated strong component $\alpha_3$ with vertices $u_1, u_2, u_3, v_1, v_2$. Taking $r_1 = u_6$, $r_2 = u_5$, $r_3^+ = u_1$, and $r_3^- = u_2$ in the proof of Theorem 2.7, it follows that

$$\mathcal{C}_5 = \begin{bmatrix} - \\ + \\ + \\ + \\ - \\ + \end{bmatrix}.$$

The last column $\mathcal{C}_6$ can be taken to be any 6 by 1 column having a $+$ and a $-$ entry, and no zeros. Let $\mathcal{C} = [\mathcal{C}_5 \mid \mathcal{C}_6]$. In order to determine a matrix $[A \mid C] \in Q([\mathcal{S} \mid \mathcal{C}])$ with a positive (left) inverse $\left[\begin{smallmatrix} B_1 \\ B_2 \end{smallmatrix}\right]$, the algorithm described in the proof of [2, Theorem 9.2.1] can be used. Then $B_1$ is a PLI of $A$, and the rows of $B_2$ are positive left nullvectors of $A$.

The next lemma is used to prove Theorem 2.10, in which square and nonsquare cases are combined.

Lemma 2.9. [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $n \geq 2$ [illegible] $\mathcal{T}$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible]

(i) $\mathcal{S}$ [illegible] $\mathcal{T}$ [illegible]

(ii) $\mathcal{S}$ [illegible] $\mathcal{T}$ [illegible]

[illegible]. Without loss of generality, assume that $\mathcal{S} = \left[\begin{smallmatrix}\mathcal{T}\\ O\end{smallmatrix}\right]$. The proof of (i) follows immediately from the definition of strong Hall.

To prove (ii), suppose first that $\mathcal{S}$ allows a PLI. Let $A_1 \in Q(\mathcal{T})$ and $A = \left[\begin{smallmatrix}A_1\\ O\end{smallmatrix}\right] \in Q(\mathcal{S})$ have $B = [B_1\ B_2]$ as a PLI. Then $B_1 A_1 = I_n$ and hence $\mathcal{T}$ allows a PLI. Next, suppose that $\mathcal{T}$ allows a PLI. Let $A_1 \in Q(\mathcal{T})$ have $B_1$ as a PLI. With $J$ denoting the all 1's matrix, it is easily verified that $B = [B_1\ J]$ is a PLI for $A = \left[\begin{smallmatrix}A_1\\ O\end{smallmatrix}\right] \in Q(\mathcal{S})$. Hence, $\mathcal{S}$ allows a PLI. The nonnegative case can be shown by a similar argument to that above. ☐

Theorem 2.10. [illegible] $m \geq n \geq 2$ [illegible] $m$ [illegible] $n$ [illegible] $\mathcal{S}$ [illegible]

(i) [illegible] $\mathcal{S}$ [illegible] $+$ [illegible] $-$ [illegible]

(ii) $\mathcal{S}$ [illegible]

(iii) [illegible] $D(\mathcal{S}_1)$ [illegible] $\mathcal{S}_1$ [illegible] $\mathcal{S}$ [illegible] $\left[\begin{smallmatrix}\mathcal{S}_1\\ O\end{smallmatrix}\right]$ [illegible] $\mathcal{S}_1$ [illegible] $n$ [illegible] $n$ [illegible]

[illegible]. For the necessity, suppose that $\mathcal{S}$ allows a PLI. Then (i) and (ii) follow from Lemmas 2.2 and 2.3, and (iii) follows from Theorem 2.4 and Lemma 2.9 (ii).

For the sufficiency, first assume $m = n$. Then $\mathcal{S}$ is permutationally equivalent to $\mathcal{S}_1$, and by Theorem 2.4 the result follows from (ii) and (iii). Next, suppose that $m > n$. If $\mathcal{S}$ has no rows of zeros, then, by Theorem 2.7, the result follows from (i) and (ii). Otherwise, without loss of generality, assume that $\mathcal{S} = \left[\begin{smallmatrix}\mathcal{T}\\ O\end{smallmatrix}\right]$, where $\mathcal{T}$ has no rows of zeros. By Lemma 2.9 (i), it follows from (ii) that $\mathcal{T}$ is strong Hall. Thus, if $\mathcal{T}$ is an $n$ by $n$ sign pattern, then (iii) and Theorem 2.4 imply that $\mathcal{T}$ allows a PLI. By Lemma 2.9 (ii), this implies that $\mathcal{S}$ allows a PLI. Otherwise, since it follows from (i) that each column of $\mathcal{T}$ has a $+$ and a $-$ entry, Theorem 2.7 implies that $\mathcal{T}$ allows a PLI. Therefore, by Lemma 2.9 (ii), $\mathcal{S}$ allows a PLI. ☐

[illegible] 2.11. For $m \geq n \geq 2$, let $\mathcal{S}$ be an $m$ by $n$ sign pattern. Then the following hold:

(i) If $\mathcal{S}$ satisfies (i), (ii), and (iii) in Theorem 2.10, then so does every superpattern of $\mathcal{S}$. Hence, if $\mathcal{S}$ allows a PLI, then every superpattern of $\mathcal{S}$ allows a PLI.

(ii) Suppose that $\mathcal{S} = \left[\begin{smallmatrix}\mathcal{S}_1\\ O\end{smallmatrix}\right]$, where $\mathcal{S}_1$ is a square sign pattern, satisfies (iii) in Theorem 2.10. Then, in contrast with Theorem 2.7 (i), there is no matrix $A = \left[\begin{smallmatrix}A_1\\ O\end{smallmatrix}\right] \in Q(\mathcal{S})$ with a PLI that also has a positive left nullvector, since the equation $[y^T\ z^T]\left[\begin{smallmatrix}A_1\\ O\end{smallmatrix}\right] = 0$ and the fact that $A_1$ is nonsingular together imply that $y = 0$.

The following theorem gives sufficient conditions for an $m$ by $n$ sign pattern with $m > n \geq 1$ to have a realization with a PLI and a positive left nullvector.

Theorem 2.12. [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $m > n$ [illegible] $\mathcal{T}$ [illegible] $t$ [illegible] $n$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$

(i) [illegible] $n = 1$ [illegible] $\mathcal{T}$ [illegible] $+$ [illegible] $-$ [illegible] $Q(\mathcal{S})$ [illegible]

(ii) [illegible] $t > n \geq 2$ [illegible] $\mathcal{T}$ [illegible] $Q(\mathcal{S})$ [illegible]

[illegible]. (i) By Proposition 2.1, a $+$ entry implies the existence of $A \in Q(\mathcal{S})$ with a

PLI. Since $A$ has a positive and a negative entry, it can be easily verified that $A$ has a positive left nullvector.

(ii) When $m = t$, the result follows by Theorem 2.7. If $m > t > n$, then Theorem 2.7 implies that there exists a matrix $A \in Q(\mathcal{T})$ with a PLI $B$ and a positive left nullvector $y^T$. Note that the positive matrix $[B \mid J]$ is a PLI and the vector $[y^T \; 1 \cdots 1]$ is a positive left nullvector of the matrix $\left[\begin{smallmatrix} A \\ O \end{smallmatrix}\right] \in Q(\mathcal{S})$. Hence, the result follows. □

**3. Nonnegative left inverses.** In this section we determine structures of nonsquare sign patterns that allow an NLI, as well as structures of NLIs.

For $m \geq n$, let $\mathcal{S}$ be an $m$ by $n$ sign pattern with a realization of rank $n$. Then, by induction, it can be shown that $\mathcal{S}$ is permutationally equivalent to

$$\begin{bmatrix} \mathcal{S}_{11} & \mathcal{S}_{12} & \cdots & \mathcal{S}_{1k} \\ O & \mathcal{S}_{22} & \cdots & \mathcal{S}_{2k} \\ \vdots & & \ddots & \vdots \\ O & \cdots & O & \mathcal{S}_{kk} \end{bmatrix}, \tag{3.1}$$

where $k \geq 1$, $\mathcal{S}_{ii}$ is a square fully indecomposable sign pattern for each $i \in \{1, \ldots, k-1\}$, and $\mathcal{S}_{kk}$ is strong Hall. Note that $\mathcal{S}$ is strong Hall if and only if $k = 1$. If $\mathcal{S}$ is an $n$ by $n$ fully indecomposable sign pattern, then $\mathcal{S}$ allows a nonnegative (left) inverse if and only if $\mathcal{S}$ allows a positive (left) inverse; see [2, Theorems 9.2.1 and 9.2.3]. In addition, [2, Theorem 9.2.6] provides a complete characterization of $n$ by $n$ partly decomposable sign patterns that allow a nonnegative (left) inverse.

[illegible] 3.1. Suppose $m > n$. Let $\mathcal{S}'$ be the square submatrix of $\mathcal{S}$ obtained by deleting the columns and rows associated with $\mathcal{S}_{kk}$. Suppose that $\mathcal{S}$ allows an NLI. Then the square sign pattern $\mathcal{S}'$ also allows an NLI. Hence, for $k = 2$, $\mathcal{S}'$ is fully indecomposable and must satisfy one of the equivalent conditions in [2, Theorem 9.2.1] (see also Theorem 2.4), and for $k \geq 3$, $\mathcal{S}'$ is partly decomposable and must satisfy the conditions in [2, Theorem 9.2.6]. Furthermore, by an argument similar to that in the proof of Lemma 2.3, it is easily verified that an NLI $B$ of a matrix in $Q(\mathcal{S})$ has the block form $B = [B_{ij}]$ with $1 \leq i, j \leq k$ and the $(i, j)$-block $B_{ij} = O$ whenever $i > j$. Thus, it follows that the strong Hall sign pattern $\mathcal{S}_{kk}$ also allows an NLI.

We now investigate various necessary and/or sufficient conditions for a strong Hall nonsquare sign pattern to allow an NLI. We first consider strong Hall sign patterns with a + and a − entry in each column.

PROPOSITION 3.2. [illegible] $m > n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $+$ [illegible] $-$ [illegible] $\mathcal{T}$ [illegible] $t$ [illegible] $n$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible] $t > n$ [illegible] $\mathcal{S}$ [illegible] $t = n$ [illegible] $\mathcal{S}$ [illegible] $D(\mathcal{T})$ [illegible]

[illegible]. The result follows directly from Theorem 2.10 and the fact that if $\mathcal{S}$ allows a PLI, then $\mathcal{S}$ allows an NLI. □

Let $\mathcal{I}_n$ denote the $n$ by $n$ sign pattern with $I_n$ as a realization, i.e., $I_n \in Q(\mathcal{I}_n)$. Clearly, $\mathcal{I}_n$ allows an NLI. Thus, in order to allow an NLI, an $m$ by $n$ sign pattern with $m \geq n$ need not have a − entry in each column as is required to allow a PLI (see Lemma 2.2), but clearly must have a + entry in each column. We first consider the case that $\mathcal{S}$ has a nonnegative column having only + or 0 entries. For ease of notation, we sometimes use $(M)_{ij}$ to denote the $(i, j)$-entry of a matrix $M$.

PROPOSITION 3.3. [illegible] $m \geq n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $\mathcal{S}$ [illegible] $m - n + 1$ [illegible]

[illegible] Let $B$ be an NLI of $A \in Q(\mathcal{S})$, and let $t$ be the number of positive entries in any nonnegative column of $A$. Without loss of generality, assume that the first column of $A$ is a nonnegative column with its first $t$ entries positive. Since $(BA)_{h1} = 0$ for each $h \in \{2, \ldots, n\}$, it follows that $B$ has the block form $B = [B_{ij}]$ with $1 \leq i, j \leq 2$, where the $(2,1)$-block $B_{21}$ is the $(n-1)$ by $t$ zero matrix. Hence, the equality $\operatorname{rank} B = n$ implies that the rank of the $(n-1)$ by $(m-t)$ matrix $B_{22}$ is $n-1$. Thus, $n-1 \leq m-t$ and the result follows. ☐

If all columns are nonnegative, then the following result gives a necessary and sufficient condition for such a sign pattern to allow an NLI.

THEOREM 3.4. [illegible] $m \geq n \geq 1$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible]

$$\begin{bmatrix} \mathcal{I}_n \\ \mathcal{T} \end{bmatrix},$$

[illegible] $\mathcal{T}$ [illegible] $(m-n)$ [illegible] $n$ [illegible]

[illegible] The case $n = 1$ follows directly from Proposition 2.1. Suppose that $n \geq 2$. For the sufficiency, assume without loss of generality that

$$\mathcal{S} = \begin{bmatrix} \mathcal{I}_n \\ \mathcal{T} \end{bmatrix}.$$

Let $T \in Q(\mathcal{T})$ and $A = \left[\begin{smallmatrix} I_n \\ T \end{smallmatrix}\right] \in Q(\mathcal{S})$. Since $[I_n \mid O]\, A = I_n$, it follows that $\mathcal{S}$ allows an NLI.

For the necessity, suppose that $\mathcal{S} = [s_{ij}]$ allows an NLI; i.e., there exist $A = [a_{ij}] \in Q(\mathcal{S})$ and an $n$ by $m$ nonnegative matrix $B = [b_{ij}]$ such that $BA = I_n$. Let $i \in \{1, \ldots, n\}$. Since $(BA)_{ii} = 1$, there exists $j_i \in \{1, \ldots, m\}$ such that $b_{ij_i} a_{j_i i} > 0$. This implies that $s_{j_i i} = +$. Also, for each $k \in \{1, \ldots, n\} \setminus \{i\}$, $(BA)_{ik} = 0$ implies that $b_{ij_i} a_{j_i k} = 0$. Thus, row $j_i$ of $\mathcal{S}$ is equal to row $i$ of $\mathcal{I}_n$. As this holds for each $i \in \{1, \ldots, n\}$, the result follows. ☐

[illegible] 3.5. Let $\mathcal{S} = \left[\begin{smallmatrix} \mathcal{I}_n \\ \mathcal{J} \end{smallmatrix}\right]$ be the $m$ by $n$ nonnegative sign pattern with $m \geq n \geq 2$, where $\mathcal{J}$ is the sign pattern with all entries positive. Then, by Theorem 3.4, $\mathcal{S}$ allows an NLI. However, in contrast with Remark 2.11 (i), Theorem 3.4 implies that no nonnegative superpattern of $\mathcal{S}$ (except $\mathcal{S}$ itself) allows an NLI.

Next, we consider sign patterns that have at least one nonnegative column and at least one column with a $+$ and a $-$ entry. We use $e_i$ to denote the $i$th column vector of an identity matrix.

THEOREM 3.6. [illegible] $m \geq n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] $p \geq 1$ [illegible] $n-p \geq 1$ [illegible] $+$ [illegible] $-$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible]

$$\begin{bmatrix} \mathcal{I}_p & \mathcal{S}_{12} \\ \mathcal{S}_{21} & \mathcal{S}_{22} \\ O & \mathcal{S}_{32} \end{bmatrix}, \tag{3.2}$$

[illegible] $\mathcal{S}_{21}$ [illegible] $r$ [illegible] $p$ [illegible] $O$ [illegible] $s$ [illegible] $p$ [illegible] $s \geq 1$ [illegible] $n-p$ [illegible] $\mathcal{S}$ [illegible] $+$ [illegible] $-$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}_{21}$ [illegible]

[illegible] Without loss of generality, we may assume that the first $p$ columns of $\mathcal{S}$ are nonnegative, and that each of the last $n-p$ columns of $\mathcal{S}$ has a $+$ and a $-$ entry.

Since $\mathcal{S}$ allows an NLI, so does the $m$ by $p$ nonnegative sign pattern consisting of the first $p$ columns of $S$. Therefore, by Theorem 3.4, we may permute the rows of $\mathcal{S}$ to obtain a matrix of the form (3.2), where $S_{21}$ is a nonnegative matrix with no row of zeros, $O$ is an $s$ by $p$ zero matrix with $s \geq 0$, and each of the last $n - p$ columns has a + and a − entry.

Let $A$ be a matrix in $Q(\mathcal{S})$ that has an NLI, say $B$. Since $BA = I_n$, each of the vectors $e_1^T, \dots, e_n^T$ is a nontrivial, nonnegative linear combination of the rows of $A$. Since the first $p$ columns of $A$ are nonnegative and $n > p$, this requires that $s \geq 1$, and we conclude that $\mathcal{S}$ has the desired form.

If $\mathcal{S}_{21}$ is vacuous or has a column of zeros, then $\mathcal{S}$ has an $(m-1)$ by 1 zero submatrix. Hence $\mathcal{S}$ is not strong Hall, and the result follows by taking the contrapositive. ☐

PROPOSITION 3.7. $m \geq n \geq 2$ $\mathcal{S}$ $m$ $n$ $p \geq 1$ $n - p \geq 1$ + − $\mathcal{S}$ (3.2)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \\ O & A_{32} \end{bmatrix} \in Q(\mathcal{S})$$

$B = \left[\begin{smallmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{smallmatrix}\right]$ $A_{21}, A_{22}, B_{12}$ $B_{22}$ $\mathcal{S}$

(i) $B_{11}$ $B_{21}$ $B_{22}$

(ii) $\mathcal{S}_{32}$

(iii) $q$ $\mathcal{S}_{21}$ $q$ $B_{12}$

(iv) $B_{12}$ $B_{12}$ $\mathcal{S}_{21}^T$

Assume that $\mathcal{S}_{21}$ is not vacuous.

Since $BA = I_n$, it follows that $B_{21}A_{11} + B_{22}A_{21} = O$. Moreover, since $B_{21}$, $B_{22}$, $A_{11}$, and $A_{21}$ are nonnegative, and no row of $A_{11}$ or $A_{21}$ is all zeros, $B_{21} = O$ and $B_{22} = O$. Also, $BA = I_n$ implies that $B_{11}A_{11} + B_{12}A_{21} = I_p$. Since $B_{11}$, $B_{12}$, $A_{11}$, and $A_{21}$ are nonnegative, both $B_{11}A_{11}$ and $B_{12}A_{21}$ are diagonal matrices. Since $A_{11} \in Q(\mathcal{I}_p)$, $A_{11}$ is an invertible diagonal matrix, and hence $B_{11}$ is a diagonal matrix. Thus, (i) is proven.

Since $B_{21}$ and $B_{22}$ are zero matrices, and $BA = I_n$, $B_{23}$ is an NLI of $A_{32}$, and (ii) is proven.

Since $B_{12}A_{21}$ is a diagonal matrix and $B_{12}$ is nonnegative, the $i$th row of $B_{12}A_{21}$ is a nonnegative linear combination of the rows of $A_{21}$ (weighted by the entries of the $i$th row of $B_{12}$). As the $i$th row of $B_{12}A_{21}$ is a nonnegative multiple of $e_i^T$, and $A_{21}$ is a nonnegative matrix with no row of zeros, it follows that if the $(i, j)$-entry of $B_{12}$ is nonzero, then the $j$th row of $A_{21}$ is a multiple of $e_i^T$. In particular, this implies that each column of $B_{12}$ has at most one nonzero entry. If the $j$th row of $A_{21}$ has at least two positive entries, then column $j$ of $B_{12}$ is a column of zeros, proving (iii). If the $(i, j)$-entry of $B_{12}$ is nonzero, then the $(j, i)$-entry of $A_{21}$ is nonzero, completing the proof of (iv).

If $\mathcal{S}_{21}$ is vacuous, then $A_{21}, A_{22}, B_{12}$, and $B_{22}$ are vacuous, in which case the proofs of (i) for $B_{11}, B_{21}$ and (ii) are similar, but statements (i) for $B_{22}$, (iii), and (iv) are vacuous. ☐

For $m \geq 2$, Proposition 3.2, Theorem 3.4, and the following theorem completely characterize the $m$ by 2 sign patterns that allow an NLI.

THEOREM 3.8. [illegible] $m \geq 2$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] 2 [illegible] $+$ [illegible] $-$ [illegible] $\mathcal{S}$ [illegible] $\mathcal{S}$ [illegible] $+$ [illegible] $[0\ +]$ [illegible] $\mathcal{S}$

[illegible]. Suppose that $\mathcal{S}$ allows an NLI. Then the first column of $\mathcal{S}$ also allows an NLI. Hence, Theorem 3.4 implies that the first column of $\mathcal{S}$ has a + entry. By Theorem 3.6, we may assume without loss of generality that $\mathcal{S}$ is of the form (3.2). Since $\mathcal{S}_{32}$ is a column sign pattern, Propositions 3.7 (ii) and 2.1 imply that $\mathcal{S}_{32}$ has a + entry. Hence, $[0\ +]$ is a row of $\mathcal{S}$.

For the converse, suppose that the first column of $\mathcal{S}$ has a + entry and $[0\ +]$ is a row of $\mathcal{S}$. Suppose that $[+\ -]$ is also a row of $\mathcal{S}$. Then without loss of generality, $A \in \mathcal{S}$ has the form

$$\begin{bmatrix} a & -b \\ u & v \\ 0 & c \end{bmatrix},$$

where $a, b, c > 0$, and $u$ and $v$ are $(m-2)$ by 1 vectors. It is easy to verify that

$$\begin{bmatrix} 1/a & O & b/ac \\ 0 & O & 1/c \end{bmatrix}$$

is an NLI of $A$.

Next suppose that $[+\ -]$ is not a row of $\mathcal{S}$. Then without loss of generality, $A \in \mathcal{S}$ has the form

$$\begin{bmatrix} a & b \\ u & v \\ 0 & -c \\ 0 & d \end{bmatrix},$$

where $a, c, d > 0$, $b \geq 0$, and $u$ and $v$ are $(m-3)$ by 1 vectors. It is easy to verify that

$$\begin{bmatrix} 1/a & O & b/ac & 0 \\ 0 & O & 1/c & 2/d \end{bmatrix}$$

is an NLI of $A$.

Hence, $\mathcal{S}$ allows an NLI. □

Note that the proof of Theorem 3.8 actually shows that if $\mathcal{S}$ is an $m$ by 2 matrix whose first column is nonnegative, second column has a + and a − entry, and $[0\ +]$ is one of its rows, then [illegible] matrix with sign pattern $\mathcal{S}$ has an NLI.

[illegible] 3.9. The strong Hall sign pattern

$$\mathcal{S} = \begin{bmatrix} + & - \\ + & - \\ 0 & + \end{bmatrix}$$

does not allow a PLI (by Lemma 2.2), but does allow an NLI (by Theorem 3.8) since

$$\begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 0 & 2 \end{bmatrix} = I_2.$$

In general (as noted in the introduction) an NLI is not unique. For instance,

$$\begin{bmatrix} 1/2 & 1/2 & 1/2 \\ 0 & 0 & 1/2 \end{bmatrix}$$

is another NLI of the above matrix.

In the next theorem, it is shown that if a sign pattern $\mathcal{S}$ of the form (3.2) has a $(3,2)$-block $\mathcal{S}_{32}$ that allows an NLI or PLI, then some conditions on the negative entries in $\mathcal{S}_{12}$ insure that $\mathcal{S}$ allows an NLI.

THEOREM 3.10. [illegible] $m \geq n \geq 2$ [illegible] $\mathcal{S}$ [illegible] $m$ [illegible] $n$ [illegible] (3.2) [illegible] $p \geq 1$ $n-p \geq 1$ [illegible] $\mathcal{S}_{21}$ $\mathcal{S}_{22}$ [illegible]

(i) [illegible] $\mathcal{S}_{32}$ [illegible] $\mathcal{S}_{12}$ [illegible] 0 [illegible] $-$ [illegible] $\mathcal{S}$ [illegible]

(ii) [illegible] $\mathcal{S}_{32}$ [illegible] $\mathcal{S}_{12}$ [illegible] $-$ [illegible] $\mathcal{S}$ [illegible]

[illegible]. (i) Let

$$A = \begin{bmatrix} I_p & A_{12} \\ A_{21} & A_{22} \\ O & A_{32} \end{bmatrix} \in Q(\mathcal{S}), \tag{3.3}$$

where $-A_{12} \geq 0$ and $A_{32}$ has $B_{23}$ as an NLI. Let

$$B = \begin{bmatrix} I_p & O & B_{13} \\ O & O & B_{23} \end{bmatrix} \tag{3.4}$$

with $B_{13} = -A_{12}B_{23}$, which is a nonnegative matrix. Then $B \geq 0$, $BA = I_n$, and hence the result follows.

(ii) Let $A \in Q(\mathcal{S})$ be of the form (3.3) and let $B$ be of the form (3.4). If $B_{23}$ is a PLI of $A_{32}$ and $B_{13} = -A_{12}B_{23}$, then $B_{13} > 0$, provided that the negative entries of $A_{12}$ are sufficiently large in magnitude, and $BA = I_n$ as required. ☐

## 4. Concluding remarks.

In section 3, we have characterized nonnegative sign patterns, strong Hall sign patterns with each column having a $+$ and a $-$ entry, and $m$ by 2 sign patterns that allow an NLI. For other cases, we have given some necessary or sufficient conditions for $\mathcal{S}$ to allow an NLI. A characterization for the blocks of the last column of a sign pattern $\mathcal{S}$ of the form (3.1) with $k \geq 2$ that allows an NLI remains open. We conclude by showing (in Theorem 4.2) that some conditions on the submatrix $\mathcal{S}_{kk}$ of a sign pattern $\mathcal{S}$ of the form (3.1) with $k \geq 2$ insure that $\mathcal{S}$ allows an NLI for arbitrary $\mathcal{S}_{1k}, \ldots, \mathcal{S}_{k-1,k}$.

Let $\mathcal{S}$ allow a PLI and $A \in Q(\mathcal{S})$. The following proposition, which is used to prove Theorem 4.2, describes a relation between a PLI of $A$ and the qualitative behavior of solutions of $x^TA = b^T$. The latter equation is given in the introduction as motivation for studying PLIs and NLIs.

PROPOSITION 4.1. [illegible] $m \geq n$ [illegible] $A$ [illegible] $m$ [illegible] $n$ [illegible] $A$ [illegible] $n$ [illegible] 1 [illegible] $b \geq 0$ [illegible] $m$ [illegible] 1 [illegible] $x > 0$ [illegible] $x^TA = b^T$

[illegible]. Suppose that an $n$ by $m$ matrix $B > 0$ is a PLI of $A$. For an $n$ by 1 nonzero vector $b \geq 0$, it is clear that $(b^TB)A = b^T$ and $b^TB > 0$. Hence, the result follows.

Next, suppose that for each $n$ by 1 nonzero vector $b \geq 0$ there exists an $m$ by 1 vector $x > 0$ satisfying $x^TA = b^T$. Take $b$ to be the $i$th column $e_i$ of $I_n$ and let $x_i > 0$

be a solution of $x^T A = e_i^T$. Then the matrix

$$B = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

is a PLI of $A$. □

THEOREM 4.2. [illegible] $m > s \geq 1$ [illegible] $n > t \geq 1$ [illegible] $m > n$ [illegible] $\mathcal{S}_{11}$ [illegible] $s$ [illegible] $t$ [illegible] $\mathcal{S}_{22}$ [illegible] $(m-s)$ [illegible] $(n-t)$ [illegible] $n-t=1$ [illegible] $\mathcal{S}_{22}$ [illegible] $-$ [illegible] $n-t \geq 2$ [illegible] $\mathcal{S}_{22}$ [illegible] $\left[\begin{smallmatrix}\mathcal{T}\\ O\end{smallmatrix}\right]$ [illegible] $\mathcal{T}$ [illegible] $s$ [illegible] $(n-t)$ [illegible] $\mathcal{S}_{12}$ [illegible] $\mathcal{S} = \left[\begin{smallmatrix}\mathcal{S}_{11} & \mathcal{S}_{12}\\ O & \mathcal{S}_{22}\end{smallmatrix}\right]$ [illegible]

[illegible] Let $A_{11}$ be a matrix in $Q(\mathcal{S}_{11})$ with $B_{11}$ as an NLI. By Theorem 2.12, there exists $A_{22} \in Q(\mathcal{S}_{22})$ that has a PLI $B_{22}$ and a positive left nullvector $y^T$. Let $A_{12} \in Q(\mathcal{S}_{12})$. Then $A_{12}$ can be written as $A_{12} = V_1 - V_2$, where $V_1, V_2 \geq 0$ and the entrywise (Hadamard) product $V_1 \circ V_2 = O$. Let $v_i^T \geq 0$ for $1 \leq i \leq s$ denote row $i$ of $V_1$. If $v_i \neq 0$, then by Proposition 4.1 there exists an $(m-s)$ by 1 vector $x_i > 0$ such that $x_i^T A_{22} = v_i^T$. If $v_i = 0$, then $x_i^T A_{22} = v_i^T = 0$ when $x_i^T = y^T$. Thus, $K_1 = [x_1, \ldots, x_s]^T > 0$ and $K_1 A_{22} = V_1$. Similarly, there exists $K_2 > 0$ such that $K_2 A_{22} = V_2$.

Let $A_{12}(\epsilon) = \epsilon V_1 - V_2 = (\epsilon K_1 - K_2) A_{22}$ for a sufficiently small $\epsilon > 0$ such that $K_2 - \epsilon K_1 > 0$. Note that $V_1 \circ V_2 = O$ implies that $A_{12}(\epsilon) \in Q(\mathcal{S}_{12})$. Let $B_{12} = B_{11}(K_2 - \epsilon K_1)$. Since $K_2 - \epsilon K_1 > 0$ and $B_{11} \geq 0$ with no rows of zeros, it follows that $B_{12} > 0$. It can be easily verified that $\left[\begin{smallmatrix} B_{11} & B_{12} \\ O & B_{22}\end{smallmatrix}\right]\left[\begin{smallmatrix} A_{11} & A_{12}(\epsilon) \\ O & A_{22}\end{smallmatrix}\right] = I_n$. Hence, the result follows. □

[illegible] 4.3. Take $\mathcal{S}_{11}$ and $\mathcal{S}_{22}$ in Theorem 4.2 to be $\mathcal{S}'$ in Remark 3.1 and $\mathcal{S}_{kk}$ in the form (3.1) with $k \geq 2$, respectively. Then the conditions on $\mathcal{S}_{kk}$ in Theorem 4.2 insure that the sign pattern $\mathcal{S}$ of the form (3.1) with $k \geq 2$ allows an NLI for arbitrary $\mathcal{S}_{1k}, \ldots, \mathcal{S}_{k-1,k}$.

**Acknowledgment.** The authors thank the referee for comments that improved the readability of the paper.

## REFERENCES

[1] A. BERMAN AND B. D. SAUNDERS, *Matrices with zero line sums and maximal rank*, Linear Algebra Appl., 40 (1981), pp. 229–235.

[2] R. A. BRUALDI AND B. L. SHADER, *Matrices of Sign-Solvable Linear Systems*, Cambridge University Press, Cambridge, UK, 1995.

[3] R. A. BRUALDI AND B. L. SHADER, *Strong Hall matrices*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 359–365.

[4] M. FIEDLER AND R. GRONE, *Characterizations of sign patterns of inverse positive matrices*, Linear Algebra Appl., 40 (1981), pp. 237–245.

[5] T. FUJIMOTO AND R. R. RANADE, *Two characterizations of inverse-positive matrices: The Hawkins-Simon Condition and the Le Chatelier-Braun Principle*, Electron. J. Linear Algebra, 11 (2004), pp. 59–65.

[6] C. R. JOHNSON, F. T. LEIGHTON, AND H. A. ROBINSON, *Sign patterns of inverse positive matrices*, Linear Algebra Appl., 24 (1979), pp. 75–83.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 566–585

# FAST QR EIGENVALUE ALGORITHMS FOR HESSENBERG MATRICES WHICH ARE RANK-ONE PERTURBATIONS OF UNITARY MATRICES*

D. A. BINI[†], Y. EIDELMAN[‡], L. GEMIGNANI[†], AND I. GOHBERG[‡]

**Abstract.** Let $\mathcal{H}_n \subset \mathbb{C}^{n\times n}$ be the class of $n \times n$ Hessenberg matrices $A$ which are rank-one modifications of a unitary matrix, that is, $A = H + \boldsymbol{u}\boldsymbol{w}^H$, where $H$ is unitary and $\mathbf{u}, \mathbf{w} \in \mathbb{C}^n$. The class $\mathcal{H}_n$ includes three well-known subclasses: unitary Hessenberg matrices, companion (Frobenius) matrices, and fellow matrices. The paper presents some novel fast adaptations of the shifted QR algorithm for computing the eigenvalues of a matrix $A \in \mathcal{H}_n$ where each step can be performed with $O(n)$ flops and $O(n)$ memory space. Numerical experiments confirm the effectiveness and the robustness of these methods.

**Key words.** Hessenberg matrices, rank-one perturbations, unitary matrices, companion matrices, quasiseparable matrices, $QR$ iteration, eigenvalue computation, complexity

**AMS subject classifications.** 65F15, 65H17

**DOI.** 10.1137/050627563

## 1. Introduction.

The subject of this paper is the efficient computation of the eigenvalues of certain upper Hessenberg matrices $A \in \mathbb{C}^{n\times n}$ of the form

$$A = H + \boldsymbol{u}\boldsymbol{w}^H, \tag{1.1}$$

where $H \in \mathbb{C}^{n\times n}$ is unitary and $\boldsymbol{u}, \boldsymbol{w} \in \mathbb{C}^n$. Let $\mathcal{H}_n$ denote this class of matrices. If $A \in \mathcal{H}_n$ is invertible, then from the Sherman–Morrison formula one has

$$A^{-1} = H^H - \sigma H^H \boldsymbol{u}\boldsymbol{w}^H H^H, \quad \sigma = (1 + \boldsymbol{w}^H H^H \boldsymbol{u})^{-1} \in \mathbb{C},$$

which can be combined with (1.1) to give

$$A - A^{-H} = \boldsymbol{u}\boldsymbol{w}^H + \bar{\sigma} H \boldsymbol{w}\boldsymbol{u}^H H. \tag{1.2}$$

The shifted QR algorithm with shifts $\alpha_k \in \mathbb{C}$, $k = 0, 1, \dots,$

$$\begin{aligned} A_0 &= A, \\ A_k - \alpha_k I_n &= Q_k R_k, \\ A_{k+1} &:= R_k Q_k + \alpha_k I_n, \end{aligned} \tag{1.3}$$

is a well-known standard method for computing the eigenvalues of a dense matrix $A$ [13, 17, 25]. For simplicity, we refer to (1.3) as the QR iteration. A careful exploitation of (1.2) says that each matrix $A_k$, $k = 0, 1, \dots,$ generated by the QR iteration (1.3) applied to $A_0 = A \in \mathcal{H}_n$ can be represented by $O(n)$ parameters. Therefore, our main goal is to find fast and numerically robust adaptations of the customary QR iteration (1.3) for an input matrix $A_0 \in \mathcal{H}_n$ which require $O(n)$ floating point operations (flops) per iteration and $O(n)$ memory space.

---

*Received by the editors March 25, 2005; accepted for publication (in revised form) by P. Van Dooren November 15, 2006; published electronically April 20, 2007.
http://www.siam.org/journals/simax/29-2/62756.html

[†]Dipartimento di Matematica, Università di Pisa, Largo Bruno Pontecorvo 5, 56127 Pisa, Italy (bini@dm.unipi.it, gemignan@dm.unipi.it). The work of these authors was partially supported by MIUR grant 2002014121 and by GNCS-INDAM.

[‡]School of Mathematical Sciences, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel (eideyu@post.tau.ac.il, gohberg@post.tau.ac.il).

**1.1. The class $\mathcal{H}_n$.** The class $\mathcal{H}_n$ includes three well-known subclasses: unitary Hessenberg matrices, companion (Frobenius) matrices, and *fellow* matrices.

The interest on eigenvalues computation for unitary Hessenberg matrices is motivated by several applications such as signal processing [16] and least squares approximation by trigonometric polynomials [3, 18]. A fast QR iteration for unitary Hessenberg matrices was first presented in [14]. The algorithm relies upon the Schur representation of a unitary Hessenberg matrix as product of (modified) Givens rotations [12]. Suitable variants of the algorithm of [14] are developed in [15, 24, 23], while the potential stability problems encountered by these algorithms are analyzed in [19].

The companion (Frobenius) matrix associated with the $n$-degree polynomial $f(z) = \sum_{i=0}^{n} f_i z^i$, $f_n \neq 0$, is defined by $F = C + \boldsymbol{f}\boldsymbol{e}_n^T$, where $C = (c_{i,j})$ is the generator of the *circulant* matrix algebra such that $c_{i,j} = 1$ for $i - j = 1 \mod n$, $c_{i,j} = 0$, otherwise $\boldsymbol{e}_n$ stands for the $n$th column of the $n \times n$ identity matrix $I_n$ and $\boldsymbol{f}^T = [-f_0/f_n - 1, -f_1/f_n, \ldots, -f_{n-1}/f_n]$. Matrix methods based on the QR iteration applied to a companion matrix are customary for polynomial root-finding: in fact, the MATLAB[1] command `roots` relies on this approach.

Matrices of the more general form $F = H + \boldsymbol{u}\boldsymbol{e}_n^T$, where $H$ is unitary Hessenberg, are referred to as fellow matrices in [6]. The root-finding problem for a polynomial expressed as a linear combination of Szegö polynomials is reduced to the solution of an eigenvalue problem for a fellow matrix (see [1] and also [2]).

From the Schur parameterization it immediately follows that any unitary Hessenberg matrix $H$ has low rank submatrices in its off-diagonal blocks. The property extends to companion and fellow matrices and is a manifestation of their *quasiseparable* structure, i.e., of the fact that all the submatrices of $H$ which do not contain the diagonal have small rank. The class of (block) quasiseparable matrices was introduced and studied in the papers [8, 9, 10, 11], in the monograph [7], and in [21]. We refer the reader to section 2 for a formal definition of quasiseparable matrices.

By using (1.2), it is also found that the quasiseparable structure of an invertible matrix $A \in \mathcal{H}_n$ is inherited by each matrix $A_k$ generated by the QR iteration (1.3) applied to $A_0 = A$. A fast QR iteration for nonsingular companion matrices based on the formula (1.2) has been recently designed in [5]. The algorithm is computationally appealing but numerically unstable due to the appearance of the inverse matrix in the formula (1.2).

**1.2. Summary of the results.** In this paper we present a unified approach to the computation of eigenvalues of unitary Hessenberg, companion, and fellow matrices by using fast, stable adaptations of the structured QR iteration. The new algorithms do not suffer the instabilities and/or the computational problems of previous works on fast QR algorithms for fellow matrices [6] and companion matrices [5]; for unitary Hessenberg matrices the new algorithms do not have shift restrictions as for previous methods [15, 24, 23].

The key idea of our approach is to find a compact representation of the matrices $A_k$, $k = 0, 1, \ldots$, generated by the QR iteration (1.3) which does not involve quantities expressed in terms of the entries of $A_k^{-1}$. To do this we elaborate on the relation (1.1). By using $A_{k+1} = Q_k^H A_k Q_k$, one obtains that

$$A_{k+1} = H_{k+1} + \boldsymbol{u}_{k+1}\boldsymbol{w}_{k+1}^H, \quad k \geq 0, \tag{1.4}$$

---

[1] MATLAB is a registered trademark of The Mathworks, Inc.

where $H_{k+1}$ is unitary and, moreover, $H_{k+1} = Q_k^H H_k Q_k$, $\boldsymbol{u}_{k+1} = Q_k^H \boldsymbol{u}_k$, and $\boldsymbol{w}_{k+1} = Q_k^H \boldsymbol{w}_k$ subjected to the initializations $H_0 = H$, $\boldsymbol{u}_0 = \boldsymbol{u}$, and $\boldsymbol{w}_0 = \boldsymbol{w}$. Since $A_{k+1}$ is Hessenberg, from (1.4) we deduce that $H_{k+1}$ has a quasiseparable structure in its strictly lower triangular part. By exploiting this property, we derive a quasiseparable representation for the whole matrix $H_{k+1}$. Combining this representation with the formula (1.4) yields the desired compact description of the matrix $A_{k+1}$ in terms of number of parameters which is linear in $n$.

The computation of the quasiseparable representation of the matrix $H_{k+1}$ in floating point arithmetic leads to some additional difficulties and sometimes in numerical tests performed with very large matrices ($n \geq 1000$) it turned out to be prone to numerical instabilities. In particular, if $fl(H_{k+1}) = \tilde{H}_{k+1} + \tilde{\Delta}_{k+1}$, where $\tilde{H}_{k+1}$ is numerically unitary, then the computation can amplify the previously accumulated error $\tilde{\Delta}_{k+1}$. A renormalization technique is used in our algorithm to overcome this drawback. The new iterate at step $k+1$ is defined as a small perturbation of $A_{k+1}$ which is both upper Hessenberg and a rank-one perturbation of a numerically unitary matrix $U_{k+1}$. In this way, the quasiseparable structure of $U_{k+1}$ and, a fortiori of the new iterate, can be computed without any amplification of previously accumulated errors. The overall computational cost of the resulting algorithm is $O(n)$ flops per iteration and $O(n)$ storage locations. Numerical experiments show that the algorithm is stable.

**1.3. Paper organization.** The paper is organized as follows. In section 2 we recall some basic properties and algorithms for quasiseparable matrices. In section 3 we first characterize the structure of a unitary matrix $H$ such that $H + \boldsymbol{u}\boldsymbol{w}^H \in \mathcal{H}_n$ for vectors $\boldsymbol{u}$ and $\boldsymbol{w}$, then we apply this result to determine a numerically suited quasiseparable representation for the matrices $A_k$ generated by the QR iteration (1.3) applied to an input matrix $A = A_0 \in \mathcal{H}_n$. In section 4, we develop fast algorithms to carry out one QR iteration in a fast and robust way using linear time and linear memory space. In section 5 we discuss practical implementations of our QR iteration algorithm and present the results of numerical experiments. Finally, the conclusion and a discussion are the subjects of section 6.

## 2. Quasiseparable structures: Definitions and fast algorithms.

For any $n \times n$ matrix $B = (b_{i,j}) \in \mathbb{C}^{n\times n}$ we adopt the MATLAB notation $\mathrm{triu}(B,p) = (t_{i,j})$ to denote the upper triangular portion of $B$ such that $t_{i,j} = b_{i,j}$ for $j - i \geq p$, and $t_{i,j} = 0$ elsewhere. Analogously, the $n \times n$ matrix $T = \mathrm{tril}(B,p)$ is formed by the lower triangular portion of $B$ such that $t_{i,j} = b_{i,j}$ for $j - i \leq p$, and $t_{i,j} = 0$ elsewhere.

A matrix $A \in \mathbb{C}^{n\times n}$ is called order $(n_L, n_U)$-[illegible] [8] if

$$n_L = \max_{1\leq k\leq n-1} \operatorname{rank} A[k+1\colon n, 1\colon k], \quad n_U = \max_{1\leq k\leq n-1} \operatorname{rank} A[1\colon k, k+1\colon n],$$

where $B[i\colon j, k\colon l]$ is the submatrix of $B$ with entries having row and column indices in the ranges $i$ through $j$ and $k$ through $l$, respectively. In the case $n_U = n_L = r$ one refers to $A$ as an order-$r$-quasiseparable matrix. A computationally important property of $n \times n$ quasiseparable matrices is that they can be represented by only $O((n_L + n_U)n)$ parameters via [illegible]. Given the set of $m \times m$ matrices $\{B_j\}_{j=2}^{n-1}$ and two positive integers $i, j$ such that $1 < i + 1 \leq j \leq n$, we define the matrix $B_{i,j}^{\times}$ as follows: $B_{i,j}^{\times} = B_{i+1} \cdots B_{j-1}$ for $n \geq j > i+1$ and $B_{j,j+1}^{\times} = I_m$ for $1 \leq j \leq n-1$. Analogously, if $j + 1 \leq i \leq n$, then ${}^{\times}B_{i,j} = B_{i-1} \cdots B_{j+1}$ for $n \geq i > j+1$ and ${}^{\times}B_{i,i-1} = I_m$ for $2 \leq i \leq n$. Then an order $(n_L, n_U)$-[illegible] matrix $A =$

$(a_{i,j}) \in \mathbb{C}^{n\times n}$ can be represented as follows (see [7] and [8]):

$$a_{i,j} = \begin{cases} \boldsymbol{p}_i^T {}^{\times}B_{i,j}\boldsymbol{q}_j, & 1 \le j < i \le n, \\ \boldsymbol{g}_i^T C^{\times}_{i,j}\boldsymbol{h}_j, & 1 \le i < j \le n. \end{cases} \tag{2.1}$$

The diagonal entries $a_{i,i}$, $1 \le i \le n$ are arbitrary and do not satisfy any structural constraint. Here $\boldsymbol{p}_i \in \mathbb{C}^{n_L}$, $2 \le i \le n$, $\boldsymbol{q}_i \in \mathbb{C}^{n_L}$, $1 \le i \le n-1$, and $B_i \in \mathbb{C}^{n_L\times n_L}$, $2 \le i \le n-1$; these elements are said to be [illegible] of the matrix $A$. Similarly, the elements $\boldsymbol{g}_i \in \mathbb{C}^{n_U}$, $1 \le i \le n-1$, $\boldsymbol{h}_i \in \mathbb{C}^{n_U}$, $2 \le i \le n$, and $C_i \in \mathbb{C}^{n_U\times n_U}$, $2 \le i \le n-1$ are said to be [illegible] of the matrix $A$.

The quasiseparable structure provides a generalization of the [illegible] structure. If $C_i = C$, $B_i = B$, and $B^{n_L} = 0$, $C^{n_U} = 0$, then the matrix $A$ defined by (2.1) is a band matrix with upper and lower bandwidth $n_U$ and $n_L$, respectively. In particular, if $n_L = 1$ and $B_i = 0$, then ${}^{\times}B_{i,j} = \delta_{i-1,j}$ is the Kronecker delta and $A$ reduces to an upper Hessenberg matrix. Roughly speaking, the quasiseparable structure is maintained under arithmetic operations, inversion, and LU and QR factorization. More precisely, the matrices generated by these operations are still quasiseparable with generally a different order of quasiseparability. Fast $O(n)$ algorithms for performing these operations, based upon generator manipulations, have already been devised in [8, 11, 10].

For our purposes a special mention is due to the algorithm for computing a QR factorization of a (block) quasiseparable matrix $A = QR$ presented in [11] (derived there via applying the more general Dewilde-van der Veen method [7] to finite quasiseparable matrices). The algorithm first reduces the matrix $A$ into block upper Hessenberg form $S = V^HA$ and then transforms $S$ into a triangular matrix $R = U^HS$ where $U$ and $V$ are unitary. It is worth observing that in certain cases where $A$ has a very special quasiseparable structure in its lower triangular part, the triangularization process above can be modified in such a way that $Q = VU$ can be constructed in the usual manner as a product of Givens rotations. This latter representation of $Q$ is equivalent to the representation via generators obtained by multiplying the quasiseparable matrices $U$ and $V$ using the multiplication algorithm in [8]. Nevertheless, the approach based on Givens rotations is much more familiar in the numerical analysis community and will be pursued in this paper whenever possible.

## 3. The quasiseparable structure of QR iterates.

In this section we show the quasiseparable structure of the matrices $A_k$, $k = 0, 1, \ldots$, generated by the QR iteration (1.3) applied to the input Hessenberg matrix $A = A_0 \in \mathcal{H}_n$, $A = H + \boldsymbol{u}\boldsymbol{w}^H$, where $H \in \mathbb{C}^{n\times n}$ is unitary and $\boldsymbol{u}, \boldsymbol{w} \in \mathbb{C}^n$. The main result follows from a basic property of the matrix $H$ which is both unitary and a rank-one perturbation of a Hessenberg matrix. More precisely, in the next subsection we prove that any matrix $B \in \mathbb{C}^{n\times n}$ which satisfies these two properties must be an order-2-quasiseparable matrix.

### 3.1. The quasiseparable structure of certain unitary matrices.

Let $P \in \mathbb{C}^{n\times n}$ be a unitary matrix such that the matrix $B = P - \boldsymbol{v}\boldsymbol{z}^H$ is a Hessenberg matrix for two suitable vectors $\boldsymbol{v}, \boldsymbol{z} \in \mathbb{C}^n$. Under these assumptions we establish a condensed quasiseparable representation for the matrix $P$.

At first we consider in detail the case already studied in the literature of a unitary Hessenberg matrix. A unitary upper Hessenberg matrix $P$ with real nonnegative subdiagonal entries can be represented as a product of (modified) Givens rotations

[14], i.e.,

$$P = G_1(a_1)G_2(a_2)\cdots G_n(a_n),$$

$$G_j(a_j) = I_{j-1} \oplus \mathcal{G}(a_j) \oplus I_{n-j-1},\ \mathcal{G}(a_j) = \begin{bmatrix} -a_j & b_j \\ b_j & \bar{a}_j \end{bmatrix}, \quad 1 \le j \le n-1, \tag{3.1}$$

$$G_n(a_n) = I_{n-1} \oplus (-a_n), \quad b_j \ge 0, |a_j|^2 + b_j^2 = 1, |a_n| = 1.$$

The decomposition is usually referred to as the [illegible] of $P$. The $a_j$'s are the [illegible] of $P$ and the $b_j$'s are the [illegible]. From (3.1) we obtain the representation of $P = (p_{i,j})$ via generators [12]

$$p_{i,j} = \begin{cases} b_{i-1}\delta_{i-1,j}1, & 1 \le j < i \le n, \\ -\bar{a}_{i-1}b^{\times}_{i-1,j}a_j, & 1 \le i \le j \le n, \end{cases}$$

where $\delta_{i,j}$ denotes the Kronecker delta. It is worth noting that, different from the representation (2.1), here the quasiseparable structure includes also the diagonal entries. Given a unitary upper Hessenberg matrix $U \in \mathbb{C}^{n\times n}$ we can always determine a unitary diagonal matrix $D = \mathrm{diag}[\mathrm{e}^{\mathrm{i}\theta_1}, \ldots, \mathrm{e}^{\mathrm{i}\theta_n}]$ such that $P = D^H U D$ has nonnegative subdiagonal entries. In this way it is found that $U = (u_{i,j})$ admits the representation

$$u_{i,j} = \begin{cases} \bar{\psi}_{i-1}\delta_{i-1,j}1, & 1 \le j < i \le n, \\ -\bar{\phi}_{i-1}\psi^{\times}_{i-1,j}\phi_j, & 1 \le i \le j \le n, \end{cases} \tag{3.2}$$

for complex numbers $\phi_j$ and $\psi_j$ such that $|\phi_j|^2 + |\psi_j|^2 = 1$.

[illegible] 3.1. The computation of a Schur-like parameterization of a unitary Hessenberg matrix $U$ can be accomplished in a robust way by factoring $U$ in the QR form. If $U = QR$, where $Q$ is unitary and $R$ is upper triangular with nonnegative diagonal entries, then necessarily one has $R = I_n$ and $Q = U$. Moreover, we can transform $U$ to upper triangular form by using $n-1$ (modified) Givens rotations to annihilate the $n-1$ subdiagonal entries.

Relation (3.2) says that any unitary Hessenberg matrix is $(1,1)$ quasiseparable. The following results generalize this property to the case where the unitary matrix has a more general (quasiseparable) structure in its lower triangular part.

THEOREM 3.2. [illegible] $P = (p_{i,j}) \in \mathbb{C}^{n\times n}$ [illegible] $\mathrm{tril}(P,-2) = \mathrm{tril}(\boldsymbol{v}\boldsymbol{z}^H,-2)$ [illegible] $n$ [illegible] $\boldsymbol{v} = [v_1,\ldots,v_n]^T$ [illegible] $\boldsymbol{z} = [z_1,\ldots,z_n]^T$ [illegible] $P$ [illegible] $P = VSU$ [illegible] $V$ [illegible] $S$ [illegible] $U$ [illegible] $V$ [illegible] $S$ [illegible] $V^H = G_3(\tilde{a}_3)\cdots G_{n-1}(\tilde{a}_{n-1})$ [illegible] $S^H = G_{n-1}(\hat{a}_{n-1})\cdots G_2(\hat{a}_2)$ [illegible]

[illegible]. The reduction of the matrix $P$ into an upper Hessenberg form can be split in two stages. In the first stage the matrix $P$ is transformed into a matrix $\tilde{P}$ of lower bandwidth 2, i.e., $\tilde{p}_{i,j} = 0$ whenever $i > j+2$. The task is accomplished as follows. Choose the first rotation $G_{n-1}(\tilde{a}_{n-1}) = I_{n-2} \oplus \mathcal{G}(\tilde{a}_{n-1})$ to yield

$$\mathcal{G}(\tilde{a}_{n-1}) \begin{bmatrix} v_{n-1} \\ v_n \end{bmatrix} = \begin{bmatrix} \hat{v}_{n-1} \\ 0 \end{bmatrix}.$$

Similarly, choose the successive rotations $G_j(\tilde{a}_j) = I_{j-1} \oplus \mathcal{G}(\tilde{a}_j) \oplus I_{n-j+1}$, $3 \le j \le n-2$ to yield

$$\mathcal{G}(\tilde{a}_j) \begin{bmatrix} v_j \\ \hat{v}_{j+1} \end{bmatrix} = \begin{bmatrix} \hat{v}_j \\ 0 \end{bmatrix}, \quad j = n-2, \ldots, 3.$$

In the second stage the matrix $\tilde{P} = G_3(\tilde{a}_3) \cdots G_{n-1}(\tilde{a}_{n-1})P = V^H P$ is reduced to an upper Hessenberg form by premultiplication by the matrices $G_2(\hat{a}_2), \ldots, G_{n-1}(\hat{a}_{n-1})$ suitably chosen to annihilate the entries along the second subdiagonal. At the end of the process we find that the matrix $U = G_{n-1}(\hat{a}_{n-1}) \cdots G_2(\hat{a}_2)\tilde{P} = S^H \tilde{P}$ is unitary Hessenberg. □

3.3. The reduction process described in the proof can easily be completed to produce a QR factorization of the quasiseparable matrix $P$. Indeed, the unitary lower Hessenberg matrix $V$ in the previous theorem is similar to the matrix $V$ determined at the first stage of the algorithm in [11] applied for computing a QR factorization of the quasiseparable matrix $P$. At the second stage the cited algorithm reduces $\tilde{P}$ into a triangular form by annihilating its subdiagonal entries column-by-column. Differently, in the scheme above the annihilation of the subdiagonal entries of $\tilde{P}$ is performed one subdiagonal at the time.

The factorization $P = VSU$ stated in the previous theorem leads to a characterization of the entries in the strictly upper triangular part of the matrix $P$. More specifically, let $U = U^{(0)}$, $P = P^{(0)}$, and recall that $U$ can be represented in the form (3.2) for suitable elements $\phi_j$ and $\psi_j$ of modulus less than or equal to 1. The reduction process described in the proof of Theorem 3.2 proceeds as follows:

$$\begin{aligned} P = P^{(0)} &\xrightarrow{\mathcal{G}(\tilde{a}_{n-1})} P^{(1)} \xrightarrow{\mathcal{G}(\tilde{a}_{n-2})} \cdots \xrightarrow{\mathcal{G}(\tilde{a}_3)} P^{(n-3)}, \\ P^{(n-3)} = \tilde{P} &\xrightarrow{\mathcal{G}(\hat{a}_2)} \cdots \xrightarrow{\mathcal{G}(\hat{a}_{n-1})} P^{(2n-5)} = U^{(0)}, \end{aligned} \tag{3.3}$$

where at the $j$th step $P^{(j-1)}$ is premultiplied by the corresponding Givens rotation to obtain $P^{(j)}$. Let us define $U^{(s)} = P^{(2n-5-s)} = (u_{i,j}^{(s)})$ for $0 \leq s \leq 2n-5$. The desired characterization of the entries of $P$ is thus found by carrying out the scheme (3.3) in the reverse order, i.e.,

$$\begin{aligned} U = U^{(0)} &\xrightarrow{\mathcal{G}(\hat{a}_{n-1})^H} U^{(1)} \xrightarrow{\mathcal{G}(\hat{a}_{n-2})^H} \cdots \xrightarrow{\mathcal{G}(\hat{a}_2)^H} U^{(n-2)}, \\ U^{(n-2)} = \tilde{P} &\xrightarrow{\mathcal{G}(\tilde{a}_3)^H} \cdots \xrightarrow{\mathcal{G}(\tilde{a}_{n-1})^H} U^{(2n-5)} = P, \end{aligned} \tag{3.4}$$

where at the $j$th step $U^{(j-1)}$ is premultiplied by the corresponding Givens rotation to obtain $U^{(j)}$. Based on the scheme (3.4), we determine the structure of the matrix $\tilde{P} = SU$ and then the structure of the matrix $P = V\tilde{U}$.

The next result shows that the upper triangular part of the matrix $\tilde{P} = SU$ obtained in the middle of the process (3.4) is order-2-quasiseparable. Note that here the structure includes the diagonal entries. A simple proof of the rank property is based on the following argument. For any fixed integer $k$, $1 \leq k \leq n-1$, we easily find that

$$U^{(j)}[1:k, k:n] = U[1:k, k:n], \quad 0 \leq j \leq n-1-k,$$

and, hence, $\operatorname{rank}(U^{(j)}[1:k, k:n]) = 1$. At the $(n-k)$th step of (3.4) the matrix $U^{(n-k)}$ is determined as $U^{(n-k)} = G(\hat{a}_k)^H U^{(n-k-1)}$. This produces a modification in the last row of $U^{(n-k)}[1:k, k:n]$ so that $\operatorname{rank}(U^{(n-k)}[1:k, k:n]) \leq 2$. The successive transformations only perform linear combinations among the rows of the considered submatrix and, therefore, do not modify this value of the rank. By using appropriate parameterizations for the matrices involved, the proof can be turned into an algorithm to compute recursively the generators of the quasiseparable structure of $\tilde{P}$.

LEMMA 3.4. [illegible] 3.2 [illegible] $S$ [illegible]

$$S^H = G_{n-1}(\hat{a}_{n-1}) \cdots G_2(\hat{a}_2),$$

[illegible] $U$ [illegible] (3.2) [illegible] $\boldsymbol{q}_j \in \mathbb{C}^2$ $\boldsymbol{t}_j \in \mathbb{C}^2$ $1 \leq j \leq n$ [illegible] $B_j \in \mathbb{C}^{2\times 2}$ $1 \leq j \leq n-1$ [illegible]

$$(\tilde{P})_{i,j} = \boldsymbol{q}_i^T B^{\times}_{i-1,j} \boldsymbol{t}_j, \quad 1 \leq i \leq j \leq n. \tag{3.5}$$

[illegible] $B_j$ [illegible] $\boldsymbol{q}_j$ [illegible] $\boldsymbol{t}_j$ [illegible]

$$B_1 = I_2, \quad B_j = \begin{bmatrix} \psi_{j-1} & 0 \\ \bar{\hat{a}}_j \bar{\phi}_{j-1} & \hat{b}_j \end{bmatrix}, \; 2 \leq j \leq n-1; \tag{3.6}$$

$$\boldsymbol{q}_1^T = [-1, 0], \; \boldsymbol{q}_2^T = [0, 1], \quad \boldsymbol{q}_j^T = \left[-\hat{b}_{j-1}\bar{\phi}_{j-2}, \hat{a}_{j-1}\right], \quad 3 \leq j \leq n; \tag{3.7}$$

$$\boldsymbol{t}_1 = \begin{bmatrix} \phi_1 \\ 0 \end{bmatrix}, \quad \boldsymbol{t}_j = \begin{bmatrix} \psi_{j-1}\phi_j \\ u^{(n-j)}_{j,j} \end{bmatrix}, \quad 2 \leq j \leq n; \tag{3.8}$$

[illegible]

$$u^{(n-j)}_{j,j} = -\bar{\hat{a}}_j u^{(0)}_{j,j} - \hat{b}_j \bar{\hat{a}}_{j+1} \bar{\psi}_j, \quad 2 \leq j \leq n-1.$$

[illegible] The proof is by induction. The premultiplication of $U^{(n-k-1)}$ by $G(\hat{a}_k)^H$ only changes the rows $k$ and $k+1$ of $U^{(n-k-1)}$. Observe that

$$U^{(n-k-1)}[k+2:n, 1:n] = \tilde{P}[k+2:n, 1:n], \quad U^{(n-k-1)}[1:k, 1:n] = U[1:k, 1:n].$$

Let us suppose that

$$u^{(n-k-1)}_{k+1,k} = -\bar{\hat{a}}_{k+1}\bar{\psi}_k, \; u^{(n-k-1)}_{k+1,j} = [0, 1]\, B^{\times}_{k,j} \boldsymbol{t}_j, \quad k+1 \leq j \leq n,$$

where $B_j$ and $\boldsymbol{t}_j$, $k+1 \leq j \leq n$ are defined as in (3.6) and (3.8). Since the matrices $B_j$ are lower triangular, it is easily verified that the entries located on the $k$th row in the strictly upper triangular part of $U$ can be specified by

$$u^{(n-k-1)}_{k,j} = \left[-\bar{\phi}_{k-1}, 0\right] B^{\times}_{k,j} \boldsymbol{t}_j, \quad k+1 \leq j \leq n.$$

The thesis follows by observing that

$$\mathcal{G}(\hat{a}_k)^H \begin{bmatrix} -\bar{\phi}_{k-1} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} [0, 1]\, B_k \\ \boldsymbol{q}^T_{k+1} \end{bmatrix}. \qquad \square$$

The description of the upper triangular part of $\tilde{P} = U^{(n-2)}$ via generators yields a quasiseparable representation of the upper triangular part of $P = U^{(2n-5)}$ generated at the end of the process (3.4). In fact, each of the remaining steps $U^{(j)} \to U^{(j+1)}$ in (3.4), $n-2 \leq j \leq 2n-6$ amounts to performing a linear combination between two consecutive rows of $U^{(j)}$ starting from the third row without modifying the general rank structure of the matrix. Taking a look at the Givens transformations performed, we arrive at the following theorem.

THEOREM 3.5. [illegible] 3.2 [illegible] $V$ [illegible] $S$ [illegible]

$$V^H = G_3(\tilde{a}_3)\cdots G_{n-1}(\tilde{a}_{n-1}), \quad S^H = G_{n-1}(\hat{a}_{n-1})\cdots G_2(\hat{a}_2),$$

[illegible] $U$ [illegible] (3.2) [illegible] $\tilde{\boldsymbol{q}}_j \in \mathbb{C}^2$ $1 \le j \le n-1$ [illegible] $\boldsymbol{t}_j \in \mathbb{C}^2$ $2 \le j \le n$ [illegible] $B_j \in \mathbb{C}^{2\times 2}$ $2 \le j \le n-1$ [illegible]

$$(P)_{i,j} = \tilde{\boldsymbol{q}}_i^T B_{i,j}^{\times} \boldsymbol{t}_j, \quad 1 \le i < j \le n. \tag{3.9}$$

[illegible] $B_j$ [illegible] $\boldsymbol{t}_j$ [illegible] (3.6) [illegible] (3.8) [illegible] $\tilde{\boldsymbol{q}}_j$ [illegible]

$$\begin{aligned} \tilde{\boldsymbol{q}}_j^T &= -\bar{\tilde{a}}_j \hat{\boldsymbol{q}}_j^T B_j + \tilde{b}_j \boldsymbol{q}_{j+1}^T, \\ \hat{\boldsymbol{q}}_{j+1}^T &= \tilde{b}_j \hat{\boldsymbol{q}}_j^T B_j + \tilde{a}_j \boldsymbol{q}_{j+1}^T, \quad j = 3 \ldots n-1, \end{aligned}$$

[illegible] $\boldsymbol{q}_j$ [illegible] (3.7) $\tilde{\boldsymbol{q}}_1^T = \boldsymbol{q}_1^T B_1$ $\tilde{\boldsymbol{q}}_2^T = \boldsymbol{q}_2^T B_2$ [illegible] $\hat{\boldsymbol{q}}_3^T = \boldsymbol{q}_3^T$

[illegible] The proof is by induction. The matrices $U^{(j+n-5)}$ and $U^{(j+n-4)} = G(\tilde{a}_j)U^{(j+n-5)}$, $3 \le j \le n-1$ only differ in the rows $j$ and $j+1$. Let us assume that

$$u_{j,k}^{(j+n-5)} = \hat{\boldsymbol{q}}_j^T B_{j-1,k}^{\times} \boldsymbol{t}_k, \quad j \le k \le n,$$

where $B_k$ and $\boldsymbol{t}_k$, $j \le k \le n$ are defined as in Lemma 3.4 and $\hat{\boldsymbol{q}}_j^T$ is generated after $j-3$ iterations of the two-step procedure above. Since we have

$$u_{j+1,k}^{(j+n-5)} = \boldsymbol{q}_{j+1}^T B_{j,k}^{\times} \boldsymbol{t}_k, \quad j+1 \le k \le n,$$

we find

$$\mathcal{G}(\tilde{a}_j)^H \begin{bmatrix} \hat{\boldsymbol{q}}_j^T B_j \\ \boldsymbol{q}_{j+1}^T \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{q}}_j^T \\ \hat{\boldsymbol{q}}_{j+1}^T \end{bmatrix},$$

and, therefore,

$$u_{j,k}^{(j+n-4)} = \tilde{\boldsymbol{q}}_j^T B_{j,k}^{\times} \boldsymbol{t}_k, \quad j < k \le n,$$

and

$$u_{j+1,k}^{(j+n-4)} = \hat{\boldsymbol{q}}_{j+1}^T B_{j,k}^{\times} \boldsymbol{t}_k, \quad j+1 \le k \le n. \qquad \square$$

### 3.2. The main result.

We are now ready to prove the main result of this section concerning the quasiseparable structure of the matrices $A_k = (a_{i,j}^{(k)})$, $k = 0, 1, \ldots$, generated by the QR iteration (1.3) applied to the input Hessenberg matrix $A = A_0 \in \mathcal{H}_n$, $A = H + \boldsymbol{u}\boldsymbol{w}^H$, where $H \in \mathbb{C}^{n\times n}$ is unitary and $\boldsymbol{u}, \boldsymbol{w} \in \mathbb{C}^n$. Since $A = A_0$ is upper Hessenberg each iterate $A_k$ has the same Hessenberg shape as $A_0$. For notational convenience we also denote $\beta_j^{(k)} = a_{j+1,j}^{(k)}$, $1 \le j \le n-1$, the subdiagonal entries of $A_k$. Since $A_{k+1} = Q_k^H A_k Q_k$ for any $k \ge 0$, from (1.1) we obtain

$$\begin{aligned} A_{k+1} &= H_{k+1} + \boldsymbol{u}_{k+1}\boldsymbol{w}_{k+1}^H, \\ H_{k+1} &= Q_k^H H_k Q_k, \\ \boldsymbol{u}_{k+1} &= Q_k^H \boldsymbol{u}_k, \ \boldsymbol{w}_{k+1} = Q_k^H \boldsymbol{w}_k, \end{aligned} \tag{3.10}$$

subjected to the initializations $H = H_0 \in \mathbb{C}^{n\times n}$, $\boldsymbol{u}_0 = \boldsymbol{u} = \left[u_1^{(0)}, \ldots, u_n^{(0)}\right]^T$, and $\boldsymbol{w}_0 = \boldsymbol{w} = \left[w_1^{(0)}, \ldots, w_n^{(0)}\right]^T$.

Based on the relations (3.10), we derive a quasiseparable representation for the matrices $A_k$, $k \geq 0$. Because of the Hessenberg form of $A_k$ from (3.10), we obtain that each matrix $H_k$ meets the assumption of Theorem 3.2. Therefore, by substituting the representation provided by Theorem 3.5 into the equations (3.10) we obtain the following theorem.

THEOREM 3.6. [illegible] $A_k = (a_{i,j}^{(k)})$ [illegible] (1.3) [illegible] $A_0 = H_0 + \boldsymbol{u}_0\boldsymbol{w}_0^H \in \mathcal{H}_n$ [illegible] $\boldsymbol{q}_j^{(k)} \in \mathbb{C}^2$ $1 \leq j \leq n-1$ [illegible] $\boldsymbol{t}_j^{(k)} \in \mathbb{C}^2$ $2 \leq j \leq n$ [illegible] $B_j^{(k)} \in \mathbb{C}^{2\times 2}$ $2 \leq j \leq n-1$ [illegible]

$$a_{i,j}^{(k)} = \boldsymbol{q}_i^{(k)^T} B_{i,j}^{(k)^\times} \boldsymbol{t}_j^{(k)} + u_i^{(k)} \bar{w}_j^{(k)}, \qquad 1 \leq i < j \leq n. \tag{3.11}$$

[illegible] 3.7. Quasiseparable representations where the parameters are allowed to vary greatly in magnitude can be prone to numerical instabilities [22]. This raises the important question of finding a well-conditioned parameterization among all the possible sets of generators of a given quasiseparable matrix. It is instructive to discuss the robustness of (3.9)–(3.11) compared with a different parameterization for the matrix $A_k$ used in [5] and derived from (1.2). It is immediately found that $\| \boldsymbol{t}_j \|_2 \leq \sqrt{2}$, $\| B_j \|_F \leq \sqrt{2}$ and $\| B_{i,j}^\times \|_F = O(\sqrt{j-i})$. Moreover, for the vectors $\boldsymbol{q}_j$ defined in (3.7) we have $\| \boldsymbol{q}_j \|_2 \leq 1$ and $\| \left[ B_{0,j}^{\times^T} \boldsymbol{q}_1, \ldots, B_{j-1,j}^{\times^T} \boldsymbol{q}_j \right]^T \|_F = O(j)$. Since the Frobenius norm is invariant under multiplication by a unitary matrix, it follows that the vectors $\tilde{\boldsymbol{q}}_j$ and $\hat{\boldsymbol{q}}_j$ generated by the coupled recurrences in Theorem 3.5 are such that $\| \tilde{\boldsymbol{q}}_j \|_2 = O(j)$ and $\| \hat{\boldsymbol{q}}_j \|_2 = O(j)$. Therefore we may conclude that the parameters used in the representation (3.9) of the strictly upper triangular part of $P$ can be bounded from above in magnitude by $O(n)$ and, hence, the parameterization (3.11) is numerically robust. In contrast to this there are cases where the formula (1.2) would generate large cancellation errors and would not be robust. To see this, let us suppose that $A = A_0$ is ill conditioned; more precisely, assume that $A$ has a fixed norm but $A^{-1}$ has an arbitrarily large norm. Since $A_k$ is unitarily similar to $A$, we have $\| A_k \|_2 = \| A \|_2$. The algorithm in [5] employs the formula (1.2) to obtain a representation of the strictly upper triangular part of $A_k$ as the sum of the strictly upper triangular parts of three rank-one matrices. The first matrix is $\boldsymbol{u}_k\boldsymbol{w}_k^H$ which is uniformly bounded. The other two rank-one matrices can have entries with arbitrarily large moduli which must combine together so that the cumulative contribution has bounded norm.

## 4. The structured QR iteration.

In this section we describe a fast adaptation of the QR iteration (1.3) for an input matrix $A = A_0 \in \mathcal{H}_n$. By exploiting the quasiseparable structure (3.11) of each iterate $A_k$, we obtain linear time and linear memory space per iteration. Given a condensed representation of the form (3.11) for the matrix $A_k$ our algorithm computes a similar representation for the matrix $A_{k+1}$ defined as in (1.3). The resulting process is referred to as the [illegible]. The computation proceeds as follows:

1. Compute $Q_k$ and $R_k$ such that $A_k - \alpha_k I_n = Q_k R_k$ provides a QR factorization of the left-hand side matrix.
2. Determine the vectors $\boldsymbol{u}_{k+1} := Q_k^H \boldsymbol{u}_k$, $\boldsymbol{w}_{k+1} := Q_k^H \boldsymbol{w}_k$ and the unitary matrix $H_{k+1} := Q_k^H H_k Q_k = R_k Q_k + \alpha_k I_n - \boldsymbol{u}_{k+1}\boldsymbol{w}_{k+1}^H$.
3. Set $A_{k+1} := H_{k+1} + \boldsymbol{u}_{k+1}\boldsymbol{w}_{k+1}^H$.

In view of (3.10), we see that the computed matrix $A_{k+1}$ coincides with the one obtained after having performed one step of the shifted QR process (1.3) starting from

$A_k$. From Theorem 3.6 it follows that both $A_k$ and $A_{k+1}$ can be represented in the condensed form (3.11) by means of $O(n)$ parameters. In what follows, we will show that the same holds for all the data structures involved in the computational process above. In this way matrix operations can be reduced to manipulating $O(n)$ rather than $O(n^2)$ elements with a dramatic reduction of the operation count.

Let us first specify the parameterization used for the matrices $A_k$ and $H_k$. According to the notations introduced in the previous section, the input data at iteration $k+1$ are given by: the vector $\boldsymbol{\beta}^{(k)} = \left[\beta_1^{(k)}, \dots, \beta_{n-1}^{(k)}\right]^T \in \mathbb{C}^{n-1}$ of the subdiagonal entries of $A_k$; the vector $\boldsymbol{a}^{(k)} = \left[a_{1,1}^{(k)}, \dots, a_{n,n}^{(k)}\right]^T \in \mathbb{C}^n$ of the diagonal entries of $A_k$; the vector $\boldsymbol{h}^{(k)} = \left[h_{1,1}^{(k)}, \dots, h_{n,n}^{(k)}\right]^T \in \mathbb{C}^n$ of the diagonal entries of $H_k$; the vectors $\boldsymbol{q}_j^{(k)} \in \mathbb{C}^2$, $1 \le j \le n-1$; the vectors $\boldsymbol{t}_j^{(k)} \in \mathbb{C}^2$, $2 \le j \le n$; the lower triangular matrices $B_j^{(k)} = \begin{bmatrix} \hat{\psi}_j^{(k)} & 0 \\ \hat{\phi}_j^{(k)} & \hat{\rho}_j^{(k)} \end{bmatrix} \in \mathbb{C}^{2\times 2}$, $2 \le j \le n-1$; the vector $\boldsymbol{u}_k = \left[u_1^{(k)}, \dots, u_n^{(k)}\right]^T \in \mathbb{C}^n$; the vector $\boldsymbol{w}_k = \left[w_1^{(k)}, \dots, w_n^{(k)}\right]^T \in \mathbb{C}^n$, and the shift $\alpha_k$. The entries of the unitary matrix $H_k = (h_{i,j}^{(k)})$ are expressed in terms of these parameters as

$$
\begin{aligned}
h_{i,j}^{(k)} &= -u_i^{(k)} \bar{w}_j^{(k)} \text{ for } i-j \ge 2, \\
h_{j+1,j}^{(k)} &= \beta_j^{(k)} - u_{j+1}^{(k)} \bar{w}_j^{(k)}, \\
h_{j,j}^{(k)} &= (\boldsymbol{h}^{(k)})_j, \\
h_{i,j}^{(k)} &= \boldsymbol{q}_i^{(k)T} B_{i,j}^{(k)\times} \boldsymbol{t}_j^{(k)} \text{ for } j-i \ge 1.
\end{aligned}
\tag{4.1}
$$

Analogously, the nonzero entries of the upper Hessenberg matrix $A_k$ are defined by

$$
\begin{aligned}
a_{j+1,j}^{(k)} &= \beta_j^{(k)}, \\
a_{j,j}^{(k)} &= (\boldsymbol{a}^{(k)})_j, \\
a_{i,j}^{(k)} &= \boldsymbol{q}_i^{(k)T} B_{i,j}^{(k)\times} \boldsymbol{t}_j^{(k)} + u_i^{(k)} \bar{w}_j^{(k)} \text{ for } j-i \ge 1.
\end{aligned}
\tag{4.2}
$$

If $H_0$ is unitary Hessenberg, then at the very beginning for $k=0$ we may set $\boldsymbol{q}_j^{(0)} = \left[-\bar{\phi}_{j-1}^{(0)} \psi_j^{(0)}, 0\right]$, $\boldsymbol{t}_j^{(0)} = \left[\phi_j^{(0)}, 0\right]^T$, and $B_j^{(0)} = \begin{bmatrix} \psi_j^{(0)} & 0 \\ 0 & 0 \end{bmatrix}$, where the $\phi_j^{(0)}$'s and the $\psi_j^{(0)}$'s define the (modified) Schur parameterization for $H_0$. Otherwise, if $H_0$ is not in Hessenberg form, the entries of $\boldsymbol{q}_j^{(0)}$, $\boldsymbol{t}_j^{(0)}$, and $B_j^{(0)}$ are to be determined according to Theorem 3.5 and Remark 3.1 by performing a preliminary QR factorization of $H_0$.

In the next subsection the structure of the matrix $A_k$ is used in order to prove that the unitary factor $Q_k$ and the upper triangular factor $R_k$ can be determined in linear time. The design of an efficient method for computing the parameterization of $H_{k+1}$ given a structural representation for $A_k$, $Q_k$, and $R_k$ is the subject in subsections 4.2 and 4.3. Finally, in subsection 4.4 we give the details about the deflation strategy.

**4.1. The QR step.** In this section we describe the QR step applied to $A_k$ and obtain the structure of $R_k$ and $Q_k$. For the sake of notational simplicity we omit the superscript $(k)$ for the matrix and the vector entries and denote by $\alpha$ the shift parameter $\alpha_k$. Since $A_k - \alpha I_n$ is upper Hessenberg, the reduction to upper triangular form can be achieved by means of a sequence of Givens rotations $G_1(\hat{a}_1), \dots, G_{n-1}(\hat{a}_{n-1})$ suitably chosen to annihilate the subdiagonal entries. The triangularization process proceeds as follows:

$$A - \alpha I_n = A^{(0)} \overset{\mathcal{G}(\hat{a}_1)}{\rightarrow} A^{(1)} \overset{\mathcal{G}(\hat{a}_2)}{\rightarrow} \dots \overset{\mathcal{G}(\hat{a}_{n-1})}{\rightarrow} A^{(n-1)} = R,$$

where at the $j$th step $A^{(j-1)}$ is premultiplied by the corresponding Givens rotation $G_j(\hat{a}_j)$ to obtain $A^{(j)}$. The vectors $\boldsymbol{q}_i$ and the scalars $u_i$ are modified twice according to the scheme $\boldsymbol{q}_i \rightarrow \hat{\boldsymbol{q}}_{\boldsymbol{i}} \rightarrow \tilde{\boldsymbol{q}}_{\boldsymbol{i}}$, $u_i \rightarrow \hat{\boldsymbol{u}}_{\boldsymbol{i}} \rightarrow \tilde{\boldsymbol{u}}_{\boldsymbol{i}}$, where $\boldsymbol{q}_i$, $u_i$ are used to represent the initial matrix $A$ in (4.2), $\hat{\boldsymbol{q}}_{\boldsymbol{i}}$, $\hat{\boldsymbol{u}}_{\boldsymbol{i}}$ are intermediate quantities and $\tilde{\boldsymbol{q}}_{\boldsymbol{i}}$, $\tilde{\boldsymbol{u}}_{\boldsymbol{i}}$ are the parameters of the quasiseparable representation of the final matrix $R$.

At the first step $G_1(\hat{a}_1)$ is chosen so that the entry in position $(2,1)$ of $G_1(\hat{a}_1)A^{(0)} = A^{(1)} = (a_{i,j}^{(1)})$ is made zero. Observe that only the entries in the first two rows of $A^{(1)}$ differ from the entries of $A^{(0)}$. These entries satisfy

$$a_{1,j}^{(1)} = \tilde{\boldsymbol{q}}_{\boldsymbol{1}}{}^T B_{2,j}^{\times} \boldsymbol{t}_j + \tilde{u}_1 \bar{w}_j, \quad a_{2,j}^{(1)} = \hat{\boldsymbol{q}}_{\boldsymbol{2}}{}^T B_{2,j}^{\times} \boldsymbol{t}_j + \hat{u}_2 \bar{w}_j, \quad j > 2,$$

where

$$\mathcal{G}(\hat{a}_1) \begin{bmatrix} \boldsymbol{q_1}^T B_2 \\ \boldsymbol{q_2}^T \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{\boldsymbol{1}}{}^T \\ \hat{\boldsymbol{q}}_{\boldsymbol{2}}{}^T \end{bmatrix}, \quad \mathcal{G}(\hat{a}_1) \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \tilde{u}_1 \\ \hat{u}_2 \end{bmatrix}. \tag{4.3}$$

The remaining entries in the $2 \times 2$ leading principal submatrix of $A^{(1)}$ are given by

$$\begin{bmatrix} d_1 & g_1 \\ 0 & d_2 \end{bmatrix} = \mathcal{G}(\hat{a}_1) A^{(0)}[1:2, 1:2].$$

The values of $\tilde{\boldsymbol{q}}_{\boldsymbol{1}}{}^T$, $\tilde{u}_1$, $d_1$, and $g_1$ are not modified by the subsequent Givens rotations, while the values of $\hat{\boldsymbol{q}}_{\boldsymbol{2}}{}^T$, $\hat{u}_2$, and $d_2$ only change at the second step where the matrix $A^{(1)}$ is premultiplied by $G_2(\hat{a}_2)$. At the $j$th step, $G_j(\hat{a}_j)$ is chosen so that the entry in position $(j+1, j)$ of $A^{(j)} = G_j(\hat{a}_j)A^{(j-1)}$ is made zero. Let $A^{(j-1)}[j : j+1, j : n]$ be given by

$$\begin{bmatrix} d_j & \hat{\boldsymbol{q}}_{\boldsymbol{j}}{}^T \boldsymbol{t}_{j+1} & \hat{\boldsymbol{q}}_{\boldsymbol{j}}{}^T B_{j+1} \boldsymbol{t}_{j+2} & \cdots & \hat{\boldsymbol{q}}_{\boldsymbol{j}}{}^T B_{j,n}^{\times} \boldsymbol{t}_n \\ \beta_j & d_{j+1} & \boldsymbol{q}_{\boldsymbol{j+1}}{}^T \boldsymbol{t}_{j+1} & \cdots & \boldsymbol{q}_{\boldsymbol{j+1}}{}^T B_{j+1,n}^{\times} \boldsymbol{t}_n \end{bmatrix} + \begin{bmatrix} \hat{u}_j \\ u_{j+1} \end{bmatrix} \boldsymbol{w}^H[j:n].$$

We set

$$\mathcal{G}(\hat{a}_j) \begin{bmatrix} d_j \\ \beta_j \end{bmatrix} = \begin{bmatrix} \vartheta \\ 0 \end{bmatrix}, \quad d_j \leftarrow \vartheta,$$

$$\mathcal{G}(\hat{a}_j) \begin{bmatrix} \hat{q}_j^T \boldsymbol{t}_{j+1} \\ d_{j+1} \end{bmatrix} = \begin{bmatrix} g_j \\ \gamma \end{bmatrix}, \quad d_{j+1} \leftarrow \gamma,$$

and

$$\mathcal{G}(\hat{a}_j) \begin{bmatrix} \hat{\boldsymbol{q}}_{\boldsymbol{j}}{}^T B_{j+1} \\ \boldsymbol{q}_{\boldsymbol{j+1}}{}^T \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{q}}_{\boldsymbol{j}}{}^T \\ \hat{\boldsymbol{q}}_{\boldsymbol{j+1}}{}^T \end{bmatrix}, \quad \mathcal{G}(\hat{a}_j) \begin{bmatrix} \hat{u}_j \\ u_{j+1} \end{bmatrix} = \begin{bmatrix} \tilde{u}_j \\ \hat{u}_{j+1} \end{bmatrix}.$$

Hence, after $j$ steps the matrix $A^{(j)}[j : j+1, j : n] = \mathcal{G}(\hat{a}_j)A^{(j-1)}[j : j+1, j : n]$ is given by

$$\begin{bmatrix} d_j & g_1 & \tilde{\boldsymbol{q}}_{\boldsymbol{j}}{}^T \boldsymbol{t}_{j+2} & \cdots & \tilde{\boldsymbol{q}}_{\boldsymbol{j}}{}^T B_{j+1,n}^{\times} \boldsymbol{t}_n \\ 0 & d_{j+1} & \hat{\boldsymbol{q}}_{\boldsymbol{j+1}}{}^T \boldsymbol{t}_{j+1} & \cdots & \hat{\boldsymbol{q}}_{\boldsymbol{j+1}}{}^T B_{j+1,n}^{\times} \boldsymbol{t}_n \end{bmatrix} + \begin{bmatrix} \tilde{u}_j \\ \hat{u}_{j+1} \end{bmatrix} \boldsymbol{w}^H[j:n].$$

At the end of the entire process the upper triangular matrix $R = R_k = (r_{i,j})$ turns out to be represented as follows:

$$\begin{aligned} r_{i,i} &= d_i, \\ r_{i,i+1} &= g_i, \\ r_{i,j} &= \tilde{\boldsymbol{q}}_i^T B^{\times}_{i+1,j} \boldsymbol{t}_j + \tilde{u}_i \bar{w}_j \text{ for } j - i \geq 2. \end{aligned} \tag{4.4}$$

The parameterization involves approximately $9n$ parameters and it is computed at the cost of about $40n$ flops. From (3.1) it follows that the unitary matrix $Q = Q_k = (G_{n-1}(\hat{a}_{n-1}) \cdots G_1(\hat{a}_1))^H$ is upper Hessenberg with Schur parameters $\{\bar{\hat{a}}_j\}$, $\hat{a}_n = -1$, and complementary parameters $\{\bar{\hat{b}}_j\}$. The computation of the new iterate $A_{k+1}$ essentially reduces to evaluating the entries of the product $S = RQ$. Since both factors are quasiseparable matrices, it is found that $S$ inherits the quasiseparable structure. In principle, the generators of $S$ and, a fortiori of $A_{k+1} = S + \alpha I_n$, can be computed by means of the multiplication algorithm of [8] or by the simpler method presented in the next subsection which makes direct use of the Schur parameterization of $Q$. However, such an approach has the following noticeable drawback: We know that the strictly upper triangular part of $A_{k+1}$ admits a quasiseparable representation of order 3 but the parameterization computed by multiplying the structures of $R$ and $Q$ has order 4. To circumvent this difficulty, we devise a compression strategy based on the computation of an additional QR factorization of the unitary matrix $H = H_{k+1}$. From the order-4 representation of $A_{k+1}$ we obtain a parameterization of order $(2, 3)$ for the entries of $H$. Then we show that the computation of a QR factorization of $H$ generates a new parameterization of this matrix of order $(2, 2)$. This finally leads to the desired parameterization of minimal length for $A_{k+1}$.

**4.2. Computing nonminimal quasiseparable representations of $\boldsymbol{A_{k+1}}$ and $\boldsymbol{H_{k+1}}$.** Nonminimal quasiseparable representations for the entries of $A_{k+1}$ and $H_{k+1}$ are easily obtained from the generators of the quasiseparable matrix $S = RQ$. By using the Schur parameterization of $Q = (G_{n-1}(\hat{a}_{n-1}) \cdots G_1(\hat{a}_1))^H$ we can split the computation of the matrix product into $n - 1$ simpler steps:

$$R = R^{(0)} \overset{\mathcal{G}(\hat{a}_1)^H}{\rightarrow} R^{(1)} \overset{\mathcal{G}(\hat{a}_2)^H}{\rightarrow} \cdots \overset{\mathcal{G}(\hat{a}_{n-1})^H}{\rightarrow} R^{(n-1)} = S,$$

where at the $j$th step $R^{(j-1)}$ is postmultiplied by $G_j(\hat{a}_j)^H$ to obtain $R^{(j)}$. The scalars $\bar{w}_j$ used to represent the matrix $R$ in (4.4) are changed twice according to the scheme $\bar{w}_j \to \hat{w}_j \to \tilde{w}_j$, where $\tilde{w}_j$ are elements of the quasiseparable structure of the final matrix $S$. The superdiagonal entries $g_i$ are modified in a similar way, $g_i \to \hat{g}_i \to \tilde{g}_i$, and, then, incorporated in the quasiseparable structure of $S$ by setting

$$\tilde{g}_i = \hat{\boldsymbol{q}}_{\boldsymbol{i}}^T \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{q}}_i^T, \tilde{g}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix}.$$

At the first step we have

$$\begin{bmatrix} \tilde{d}_1 & \hat{g}_1 \\ \beta_1 & \tilde{d}_2 \end{bmatrix} = \begin{bmatrix} d_1 & g_1 \\ 0 & d_2 \end{bmatrix} \mathcal{G}(\hat{a}_1)^H, \; d_1 \leftarrow \tilde{d}_1, d_2 \leftarrow \tilde{d}_2, \; [\tilde{w}_1, \hat{w}_2] = [\bar{w}_1, \bar{w}_2]\, \mathcal{G}(\hat{a}_1)^H.$$

At the successive step $\mathcal{G}(\hat{a}_2)^H$ acts on the second and third rows of $R^{(1)}$. We can rewrite

$$R^{(1)}[1:2, 2:3] = \begin{bmatrix} \tilde{g}_1 + \tilde{u}_1 \tilde{w}_2 & \tilde{\boldsymbol{q}}_1^T \boldsymbol{t}_3 + \tilde{u}_i \bar{w}_3 \\ d_2 & g_2 \end{bmatrix}$$

as

$$R^{(1)}[1:2,2:3] = \left[\begin{array}{cc} \hat{\boldsymbol{q}}_1^T \begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix} + \tilde{u}_1\tilde{w}_2 & \hat{\boldsymbol{q}}_1^T \begin{bmatrix}\boldsymbol{t}_3\\ 0\end{bmatrix} + \tilde{u}_i\bar{w}_3 \\ d_2 & g_2 \end{array}\right], \quad \hat{\boldsymbol{q}}_1^T = \left[\tilde{\boldsymbol{q}}_1^T, \tilde{g}_1\right].$$

Hence, we obtain

$$R^{(1)}[1:3,2:3]\mathcal{G}(\hat{a}_2)^H = \left[\begin{array}{cc} \hat{\boldsymbol{q}}_1^T\tilde{\boldsymbol{z}}_2 + \tilde{u}_1\tilde{w}_2 & \hat{\boldsymbol{q}}_1^T\hat{\boldsymbol{z}}_2 + \tilde{u}_i\hat{w}_3 \\ \tilde{d}_2 & \hat{g}_2 \\ \beta_2 & \tilde{d}_3 \end{array}\right],$$

where

$$[\tilde{\boldsymbol{z}}_2, \hat{\boldsymbol{z}}_{\boldsymbol{2}}] = \begin{bmatrix}\mathbf{0} & \boldsymbol{t}_3\\ 1 & 0\end{bmatrix}\mathcal{G}(\hat{a}_2)^H,\ d_2 \leftarrow \tilde{d}_2,\ d_3 \leftarrow \tilde{d}_3,,\ [\tilde{w}_2, \hat{w}_3] = [\hat{w}_2, \bar{w}_3]\,\mathcal{G}(\hat{a}_2)^H.$$

Since

$$\hat{\boldsymbol{q}}_1^T\hat{\boldsymbol{z}}_2 = \hat{\boldsymbol{q}}_1^T\left[\begin{array}{c|c} \dfrac{B_3}{\mathbf{0}^T} & \hat{\boldsymbol{z}}_2 \end{array}\right]\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix} = \hat{\boldsymbol{q}}_1^T F_2\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix}, \quad F_2 \in \mathbb{C}^{3\times 3},$$

and $\tilde{\boldsymbol{q}}_1^T B_3\boldsymbol{t}_4 = \hat{\boldsymbol{q}}_1^T F_2\left[\begin{smallmatrix}\boldsymbol{t}_4\\ 0\end{smallmatrix}\right]$, then the process can be continued. At the $j$th step we find

$$R^{(j-1)}[1:j+1,j:j+1] = \left[\begin{array}{cc} \hat{\boldsymbol{q}}_1^T F_{1,j}^{\times}\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix} + \tilde{u}_1\hat{w}_j & \hat{\boldsymbol{q}}_1^T F_{1,j}^{\times}\begin{bmatrix}\boldsymbol{t}_{j+1}\\ 0\end{bmatrix} + \tilde{u}_1 w_{j+1} \\ \vdots & \vdots \\ \hat{\boldsymbol{q}}_{j-1}^T\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix} + \tilde{u}_1\hat{w}_j & \hat{\boldsymbol{q}}_{j-1}^T\begin{bmatrix}\boldsymbol{t}_{j+1}\\ 0\end{bmatrix} + \tilde{u}_1 w_{j+1} \\ d_j & g_j \\ 0 & d_{j+1} \end{array}\right],$$

which gives the following representation of $R^{(j)}[1:j+1,j:j+1] = R^{(j-1)}[1:j+1,j:j+1]\mathcal{G}(\hat{a}_j)^H$

$$R^{(j)}[1:j+1,j:j+1] = \left[\begin{array}{cc} \hat{\boldsymbol{q}}_1^T F_{1,j}^{\times}\tilde{\boldsymbol{z}}_j + \tilde{u}_1\tilde{w}_j & \hat{\boldsymbol{q}}_1^T F_{1,j}^{\times}\hat{\boldsymbol{z}}_j + \tilde{u}_1\hat{w}_{j+1} \\ \vdots & \vdots \\ \hat{\boldsymbol{q}}_{j-1}^T\tilde{\boldsymbol{z}}_j + \tilde{u}_1\tilde{w}_j & \hat{\boldsymbol{q}}_{j-1}^T\hat{\boldsymbol{z}}_j + \tilde{u}_1\hat{w}_{j+1} \\ \tilde{d}_j & \hat{g}_j \\ \beta_j & \tilde{d}_{j+1} \end{array}\right],$$

where $d_j \leftarrow \tilde{d}_j,\ d_{j+1} \leftarrow \tilde{d}_{j+1}$, and

$$[\tilde{\boldsymbol{z}}_j, \hat{\boldsymbol{z}}_{\boldsymbol{j}}] = \begin{bmatrix}\mathbf{0} & \boldsymbol{t}_{j+1}\\ 1 & 0\end{bmatrix}\mathcal{G}(\hat{a}_j)^H, \quad [\tilde{w}_j, \hat{w}_{j+1}] = [\hat{w}_j, \bar{w}_{j+1}]\,\mathcal{G}(\hat{a}_j)^H.$$

After $n-1$ steps the matrix $S = R^{(n-1)} = (s_{i,j})$ is given by

$$\begin{aligned} &s_{i,j} = \hat{\boldsymbol{q}}_{\boldsymbol{i}}{}^T F_{i,j}^{\times}\tilde{\boldsymbol{z}}_j + \tilde{u}_i\tilde{w}_j \text{ for } j-i \geq 1,\\ &s_{i,i} = d_i,\\ &s_{i+1,i} = \beta_i,\\ &s_{i,j} = 0 \text{ for } i-j \geq 2. \end{aligned}$$

The generators of this representation are computed at the cost of $20n$ flops. The quasiseparable representation of $S = RQ$ provides a parameterization of $H = (h_{i,j}) = S + \alpha I_n - \tilde{\boldsymbol{u}}\tilde{\boldsymbol{w}}^H$ of order $(2,3)$. On the other hand, $H$ satisfies the hypotheses of Theorem 3.2, that is, $H = H_{k+1}$ is unitary and $h_{i,j} = -\tilde{u}_i\tilde{w}_j$ for $i \geq j+2$; therefore, its strictly upper triangular part must be 2-quasiseparable. In the next subsection we describe a method to compress the given parameterization of $H$ which relies upon the properties of the QR factorization of the matrix.

**4.3. Compressing the quasiseparable representation of $\boldsymbol{H_{k+1}}$.** According to Theorem 3.2 the quasiseparable structure of $H = H_{k+1}$ can be recovered from the coefficients of the Givens rotation matrices employed in the process of computing a QR factorization of the matrix. More specifically, $H$ can be first reduced to upper Hessenberg form by applying a sequence of $2n-5$ Givens rotations suitably chosen to annihilate the entries in the left-bottom corner. The transformed matrix is still unitary and, therefore, its $QR$ factorization gives the Schur and complementary parameters of its structured representation (3.2) (compare with Remark 3.1). The parameters can finally be combined with the coefficients of the previously determined Givens rotations to define the structure of the upper triangular part of $H$.

At first the matrix $H = P^{(0)}$ is transformed into a matrix $\tilde{P}$ of lower bandwidth 2 by applying a sequence of Givens rotations $G_{n-1}(\tilde{a}_{n-1}), \dots, G_3(\tilde{a}_3)$ according to the scheme given by the first relation in (3.3). The matrix $\mathcal{G}(\tilde{a}_j)$ is determined to satisfy

$$\mathcal{G}(\tilde{a}_j)\begin{bmatrix} \tilde{u}_j \\ \hat{u}_{j+1} \end{bmatrix} = \begin{bmatrix} \hat{u}_j \\ 0 \end{bmatrix}, \quad (\hat{u}_n = \tilde{u}_n).$$

The transformation $P^{(j-1)} \to P^{(j)} = G_{n-j}(\tilde{a}_{n-j})P^{(j-1)}$ only modifies the entries in the rows $n-j$ and $n-j+1$. The updating of the quasiseparable structure in the strictly upper triangular part of $P^{(j-1)}$ is performed by means of the same techniques used in the previous subsection for the matrix $R^{(j-1)}$. At the end of the process the entries in the upper triangular part of $\tilde{P} = (\tilde{p}_{i,j})$ satisfy

$$\tilde{p}_{i,j} = \tilde{\boldsymbol{p}}_{\boldsymbol{i}}^T E^{\times}_{i-1,j}\widehat{\boldsymbol{z}}_j \text{ for } j-i \geq 0,$$

for suitable vectors $\tilde{\boldsymbol{p}}_{\boldsymbol{i}}, \widehat{\boldsymbol{z}}_i \in \mathbb{C}^4$, $1 \leq i \leq n$, and matrices $E_i \in \mathbb{C}^{4\times 4}$, $1 \leq i \leq n-1$. The computation of these generators requires approximately $20n$ flops.

At the second stage the matrix $\tilde{P}$ is reduced to a Hessenberg form $\widehat{P}$ by means of a sequence of $n-2$ Givens rotations $G_2(\hat{a}_2), \dots, G_{n-1}(\hat{a}_{n-1})$ suitably chosen to annihilate the second lower subdiagonal. The reduction is carried out according to scheme provided by the second relation in (3.3) at the overall cost of about $35n$ flops. The modifications of the quasiseparable structure in the strictly upper triangular part of $\tilde{P}$ are described in the proof of Theorem 3.5. At the end of the process the entries in the strictly upper triangular part of the unitary Hessenberg matrix $\widehat{P} = (\widehat{p}_{i,j}) = G_{n-1}(\hat{a}_{n-1})\cdots G_2(\hat{a}_2)\tilde{P}$ can be represented as follows:

$$\widehat{p}_{i,j} = \widehat{\boldsymbol{p}}_{\boldsymbol{i}}^T E^{\times}_{i,j}\widehat{\boldsymbol{z}}_j \text{ for } j-i \geq 1.$$

Finally, the matrix $\widehat{P}$ can be converted into the identity matrix of order $n$ by applying $n$ modified rotations of the form

$$G_j(a'_j) = I_{j-1} \oplus \begin{bmatrix} -a'_j & b'_j \\ \bar{b}'_j & \bar{a}'_j \end{bmatrix} \oplus I_{n-j-1}, \quad 1 \leq j \leq n-1, \quad |a'_j|^2 + |b'_j|^2 = 1,$$
$$G_n(a'_n) = I_{n-1} \oplus (-a'_n), \quad |a'_n| = 1,$$

chosen in such a way that $G_n(a'_n)G_{n-1}(a'_{n-1})\cdots G_1(a'_1)\widehat{P}$ is an upper triangular matrix with nonnegative diagonal entries. The coefficients $a'_j$ and $b'_j$, $1 \le j \le n$ are computed at the cost of $20n$ flops and define the parameters $\phi_j = \overline{a'_j}$ and $\psi_j = b'_j$ used in Theorem 3.5 to describe the structure in the strictly upper triangular part of the matrix $P$. Here we have

$$P^H = H^H = G_{n-1}(a'_{n-1})\cdots G_1(a'_1)G_{n-1}(\hat{a}_{n-1})\cdots G_2(\hat{a}_2)G_3(\tilde{a}_1)\cdots G_{n-1}(\tilde{a}_{n-1}).$$

The matrix $H = H_{k+1}$ can thus be reconstructed in the desired form using the procedure given in Theorem 3.5 at the cost of approximately $20n$ flops.

**4.4. Deflation in the QR algorithm.** When an eigenvalue has been approximated by the QR iteration with sufficiently high precision, a deflation technique is generally employed before proceeding with the computation of the remaining eigenvalues. Sometimes it also happens during the computation that one of the subdiagonal entries other than the bottom one becomes practically zero so that the eigenvalue problem can be broken into two smaller subproblems. For the sake of clarity, suppose that after $k$ iterations of the QR algorithm the matrix $A_k$ has the form

$$A_k = \begin{bmatrix} A_k^{(1)} & \star \\ \bigcirc & A_k^{(2)} \end{bmatrix},$$

where $A_k^{(1)} \in \mathbb{C}^{(n-s)\times(n-s)}$ and $A_k^{(2)} \in \mathbb{C}^{s\times s}$. Now $A_k^{(1)}$ and $A_k^{(2)}$ can be reduced into upper triangular form separately. The process is called [illegible] and continuing in this fashion, by operating on smaller and smaller matrices, we may approximate all the eigenvalues of $A$.

Let us suppose that at the subsequent iterations $j \ge k$ the QR algorithm only operates on the submatrix $A_j^{(1)}$. The matrix $H_k$ can be partitioned accordingly with $A_k$ as follows:

$$H_k = \begin{bmatrix} H_k^{(1)} & \star \\ -\boldsymbol{u}_k^{(2)}\boldsymbol{w}_k^{(1)^H} & H_k^{(2)} \end{bmatrix},$$

where $\boldsymbol{u}_k^{(2)}$ is formed from the last $s$ components of the vector $\boldsymbol{u}_k$ and $\boldsymbol{w}_k^{(1)}$ is defined by the first $n-s$ elements of $\boldsymbol{w}_k$. A unitary upper Hessenberg matrix $U_s \in \mathbb{C}^{s\times s}$ such that $U_s\boldsymbol{u}_k^{(2)} = \left[\tilde{u}_{n-s+1}, \boldsymbol{0}^T\right]^T$ can be determined at the cost of $O(n)$ flops. Then we have

$$(I_{n-s} \oplus U_s)H_k(I_{n-s} \oplus U_s)^H = \begin{bmatrix} H_k^{(1)} & \star \\ -u_{n-s+1}\boldsymbol{e}_1\boldsymbol{w}_k^{(1)^H} & \star \end{bmatrix},$$

where $\boldsymbol{e}_1$ is the first column of $I_s$. In this way the computation of the quasiseparable structure in the strictly upper triangular part of $A_j^{(1)}$, $j \ge k$, can be reduced to finding a QR factorization of the $(n-s+1)\times(n-s)$ matrix

$$\left[\begin{array}{c} H_j^{(1)} \\ \hline -u_{n-s+1}\boldsymbol{w}_j^{(1)^H} \end{array}\right]$$

instead of the whole matrix $H_j$.

**5. Experimental issues.** In this section we first discuss the behavior of the [illegible] when performed in finite precision arithmetic. Then we design a practical implementation of the algorithm and present the results of extensive numerical experiments.

**5.1. The practical algorithm.** The proposed fast variant of the QR algorithm for input matrices $A \in \mathcal{H}_n$ relies upon two basic properties: $A$ is upper Hessenberg and, moreover, $A$ is a rank-one modification of a unitary matrix. Theoretically each matrix $A_k$ generated by the QR iteration applied to $A = A_0$ inherits these two properties. In practice, due to roundoff errors, the matrix $H_k = A_k - \boldsymbol{u}_k \boldsymbol{w}_k^H$ may deviate from orthogonality. This phenomenon is also observed in the customary shifted QR iteration. The error analysis in [20] says that the upper Hessenberg matrix $fl(A_k)$ generated after $k$ iterations of the classical method satisfies $fl(A_k) = \tilde{Q}^H(H_0 + \boldsymbol{u}_0\boldsymbol{w}_0^H + \epsilon\Delta_k)\tilde{Q}$, where $\epsilon$ is the machine precision, $\tilde{Q}$ and $H_0$ are unitary, and $\| \Delta_k \|_2 = O(kn\sqrt{n} \| A_0 \|_2)$. This means that $fl(A_k) = \tilde{H}_k + \tilde{\Delta}_k + \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{w}}_k^H$, where $\tilde{H}_k$ is unitary. In the structured QR iteration an important source of error amplification is the computation of the quasiseparable structure in the strictly upper triangular part of $fl(\tilde{H}_k + \tilde{\Delta}_k) = fl(fl(A_k) - \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{w}}_k^H)$ by means of the algorithm outlined in Theorem 3.5. In light of Remark 3.7, by performing a forward error analysis of the coupled recurrences for the vectors $\tilde{\boldsymbol{q}}_j$ it is found that the norm of the absolute perturbation $\tilde{\Delta}_k$ in the input data can be amplified in the output by a factor $n^2$. The combined effect of these two mechanisms, i.e., the departure from orthogonality of $fl(H_k)$ and the amplification of its absolute perturbation in the process of reconstructing the quasiseparable structure of the upper triangular part, can produce a possible deterioration in the accuracy of the computed eigenvalues. Extensive numerical tests with large random matrices ($n \geq 1000$) confirmed this claim. Sometimes the very last computed eigenvalues were significantly less accurate than the corresponding approximations returned by the customary QR iteration.

To avoid the possible amplification of errors in the computation of the structural representation of $H_k$ we proceed as follows. Assume that $H_k$ is a small perturbation of a unitary matrix. Its QR factorization is $H_k = U_k S_k$, where $U_k$ is unitary and $S_k = I_n + \Delta$ is upper triangular. Hence, from $A_k = H_k + \boldsymbol{u}_k \boldsymbol{w}_k^H$ we obtain $A_k S_k^{-1} = U_k + \boldsymbol{u}_k \boldsymbol{w}_k^H S_k^{-1}$. The matrix $A_k S_k^{-1}$ is a small perturbation of $A_k$ which is both upper Hessenberg and a rank-one modification of the numerically unitary matrix $U_k$. Thus we replace the matrix $A_k$ by the matrix $A_k S_k^{-1}$. The latter belongs to the class $\mathcal{H}_n$ and is a correction of the matrix $A_k$.

Moreover, the quasiseparable structure of $U_k$ and, a fortiori of $A_k$, can be computed as described in Theorem 3.5 without any amplification of previously accumulated errors. Therefore, in our implementation we compute the new iterate $A_k$ as $A_k := U_k + \boldsymbol{u}_k \tilde{\boldsymbol{w}}_k^H$, where $\tilde{\boldsymbol{w}}_k^H$ is formed by the first $n-2$ elements of the corrected vector $\boldsymbol{w}_k^H S_k^{-1}$ and the last 2 elements of $\boldsymbol{w}_k^H$. Since $\boldsymbol{u}_k$ and $\boldsymbol{w}_k$ are already known, we may determine the quantities $\tilde{w}_i^{(k)}$, $1 \leq i \leq n-2$, by a direct inspection of the entries $\gamma_i^{(k)}$, $1 \leq i \leq n-2$, in the second subdiagonal of $U_k$ without explicitly computing the matrix $S_k$. We set $\tilde{w}_i^{(k)} = w_i^{(k)}(1 + \delta_i^{(k)})$ and then find $(1 + \delta_i^{(k)}) = -\gamma_i^{(k)}/(u_{i+2}^{(k)} w_i^{(k)})$. Specific rules can also be defined in the case where some coefficient is zero. The resulting process is summarized below.

1. Compute $Q_k$ and $R_k$ such that $A_k - \alpha_k I_n = Q_k R_k$ provides a QR factorization of the left-hand side matrix.
2. Determine the matrix $A_{k+1} := R_k Q_k + \alpha_k I_n$ and the vectors $\boldsymbol{u}_{k+1} := Q_k^H \boldsymbol{u}_k$, $\boldsymbol{w}_{k+1} := Q_k^H \boldsymbol{w}_k$.

3. Compute $H_{k+1} = A_{k+1} - \boldsymbol{u}_{k+1}\boldsymbol{w}_{k+1}^H$.
4. Find the unitary matrix $U_{k+1}$ such that $H_{k+1} = U_{k+1}S_{k+1}$ is a QR factorization of $H_{k+1}$ and $S_{k+1}$ is upper triangular with real positive diagonal entries.
5. Compute the vector $\tilde{\boldsymbol{w}}_{k+1}$ and set $A_{k+1} := U_{k+1} + \boldsymbol{u}_{k+1}\tilde{\boldsymbol{w}}_{k+1}^H$.

This algorithm has been implemented in MATLAB and then applied to the computation of the eigenvalues of companion matrices of both small and large size. The program is available as a MATLAB function at www.dm.unipi.it/~gemignan/ric.html/compqr.m. The results of extensive numerical experiments confirm the robustness and efficiency of the proposed approach.

Our implementation requires 180 $n + O(1)$ flops per iteration. The main program incorporates the following shifting strategy suggested in [25, p. 549]. At the beginning the shift parameter $\sigma$ is equal to zero. If $A_s = (a_{i,j}^{(s)}) \in \mathbb{C}^{n\times n}$ satisfies $|a_{n,n}^{(s-1)} - a_{n,n}^{(s)}| \leq 0.3|a_{n,n}^{(s-1)}|$, then we apply nonzero shifts by setting $\sigma_k = a_{n,n}^{(k)}$, $k = s, s+1, \ldots$. We say that $a_{n,n}^{(k)}$ provides a numerical approximation of an eigenvalue $\lambda$ of $A_0$ whenever $|\beta_n^{(k)}| \leq \mathit{eps}\ (|a_{n,n}^{(k)}| + |a_{n-1,n-1}^{(k)}|)$, where *eps* is the machine precision, i.e., $\mathit{eps} \simeq 2.2 \cdot 10^{-16}$. If this condition is fulfilled, then we set $\lambda = a_{n,n}^{(k)}$ and deflate the matrix. After nonzero shifting has begun, we check for the convergence of the last diagonal entries of the currently computed iterate $A_k$. If convergence fails to occur after 15 iterations, then at the 16th iteration we set $\sigma_k = 1.5\ (|a_{n,n}^{(k)}| + |\beta_n^{(k)}|)$ and continue with nonzero shifting. If $a_{n,n}^{(k)}$ does not converge in the next 15 iterations, then the program reports failure. In our experiments such failure has never been encountered.

**5.2. Numerical experiments.** We tested companion matrices $C$ associated with the following polynomials:

1. the scaled "Wilkinson polynomial": $p(z) = \prod_{k=1}^{n}(z - k/n)$;
2. the polynomial $p(z) = z^n - 1$ with zeros equispaced on the unit circle;
3. the monic polynomial with zeros equally spaced on the curve $z = x + \mathtt{i}\sin(\pi x)$, $-1 \leq x \leq 1$, namely $p(z) = \prod_{k=-n/2}^{n/2-1}(z - \frac{2(k+0.5)}{n-1} - \mathtt{i}\sin(\frac{2(k+0.5)}{n-1}))$;
4. the polynomial $p(z) = \sum_{j=0}^{n-1} a_j z^j + z^n$ with $a_j = \text{rand} + \mathtt{i}\ \text{rand}$, $j = 0, \ldots, n-1$, where rand is a pseudorandom number uniformly distributed in $[0, 1]$;
5. the polynomial $p(z) = \sum_{j=0}^{n-1} a_j z^j + z^n$, where $a_j = a_{1,j} \times 10^{e_{1,j}} + \mathtt{i}\ a_{2,j} \times 10^{e_{2,j}}$, $a_{i,j} = \text{rand} + \mathtt{i}\ \text{rand}$, $e_{i,j} = 10 \times (\text{rand} - 0.5) + \mathtt{i}\ 10 \times (\text{rand} - 0.5)$, $i = 1, 2$, $j = 0, \ldots, n-1$.

An estimate of the maximum error expected in the computation of the eigenvalues of $C$ by using the QR iteration without balancing is $K \cdot \mathit{eps} \cdot \max(\mathtt{condeig}(C)) \cdot \parallel C \parallel$, where $K$ actually depends on the size of $C$ and on the number of QR steps. Our test program returns the value $\mathit{est\_err} = \mathit{eps} \cdot \max(\mathtt{condeig}(C)) \cdot \parallel C \parallel_2$, where the MATLAB function `condeig` is used to approximate the eigenvalue condition numbers.

Tables 5.1, 5.2, and 5.3 show the results of our numerical experiments for the above polynomials from 1. to 3. by reporting the degree $n$, the value of *est_err* and the maximum absolute error of the computed roots. Table 5.3 also reports the average number *it* of QR iterations per eigenvalue. The total number of iterations performed to compute all the eigenvalues is less than $6n$. Our implementation is not optimized for time efficiency. However, for the sake of illustration we have compared the running times of our program to a customary implementation of the standard QR eigenvalue algorithm for Hessenberg matrices written in MATLAB. The timings for the tests of Table 5.3 are reported in Figure 5.1. These results say that our implementation is

Table 5.1

| | Scaled Wilkinson polynomial | |
|---|---|---|
| $n$ | est_err | err |
| 4 | 0.8e-13 | 6.4e-15 |
| 8 | 5.4e-10 | 2.0e-11 |
| 16 | 7.2e-02 | 5.4e-05 |

Table 5.2

| | $p(z) = z^n - 1$ | | |
|---|---|---|---|
| $n$ | est_err | err | it |
| 128 | 2.2e-16 | 5.2e-15 | 5.86 |
| 256 | 2.2e-16 | 9.1e-15 | 5.91 |
| 512 | 2.2e-16 | 1.7e-14 | 5.60 |

Table 5.3

| | $p(z) = \prod_{k=-n/2}^{n/2-1}(z - \frac{2(k+0.5)}{n-1} - \mathrm{i}\sin(\frac{2(k+0.5)}{n-1}))$ | |
|---|---|---|
| $n$ | est_err | err |
| 8 | 2.2e-14 | 8.4e-15 |
| 16 | 4.8e-11 | 2.8e-12 |
| 32 | 7.9e-04 | 9.9e-07 |

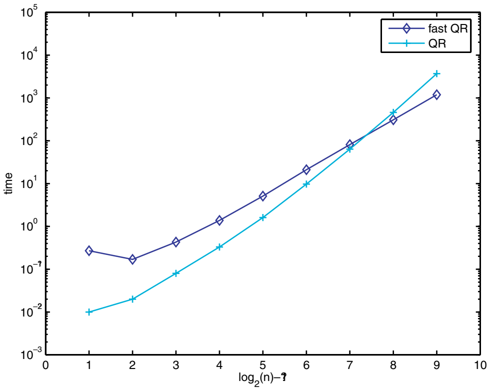


Fig. 5.1. *Timings for the tests of Table* 5.3.

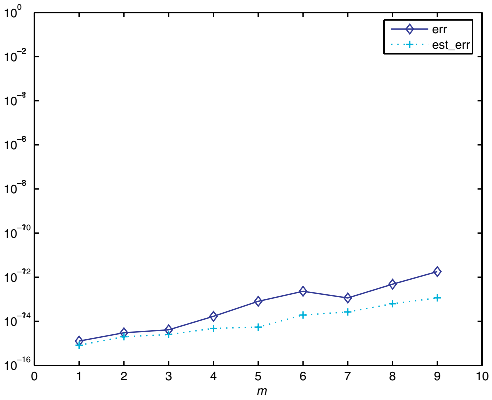


Fig. 5.2. *Random polynomials of form* 4. *of degree* $n(m) = 2^{2+m}$.

faster than the standard method for $n$ between 300 and 400 and becomes definitely faster for larger $n$.

Figures 5.2 and 5.3 cover our tests with random polynomials of the form 4.–5. of high degree. Each figure shows the *error* and the value of *est_err* for polynomials of

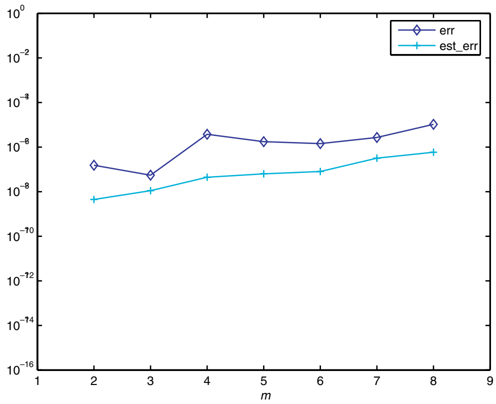


FIG. 5.3. *Random polynomials of form* 5. *of degree* $n(m) = 2^{2+m}$.

degree $n(m) = 2^{2+m}$ for $m = 1, 2, \ldots$. For each size we carried out 100 numerical experiments and reported the average values of *error* and *est_err*. The *error* measures the distance between the set of the computed eigenvalues and the set of eigenvalues returned by the function `eig` with the same input data. Let $\lambda(A)$ denote the set of eigenvalues computed by the MATLAB function `eig`. Let $\tilde{\lambda}(A)$ denote the set of eigenvalues computed by our algorithm, and define the distance between the sets $\lambda(A)$ and $\tilde{\lambda}(A)$ by $\mathrm{dist}(\lambda(A), \tilde{\lambda}(A)) = \max\{\max_{\tilde{\lambda}\in\tilde{\lambda}(A)} \parallel \tilde{\lambda} - \lambda(A) \parallel, \max_{\lambda\in\lambda(A)} \parallel \lambda - \tilde{\lambda}(A) \parallel\}$, where $\parallel \lambda - \tilde{\lambda}(A) \parallel = \min_{\tilde{\lambda}\in\tilde{\lambda}(A)} |\lambda - \tilde{\lambda}|$. We refer to this distance as the error in the eigenvalues computed by our algorithm. Therefore, we tacitly assume that the MATLAB function `eig` computes the eigenvalues exactly.

## 6. Conclusion.

In this paper we have presented a novel QR eigenvalue algorithm for a class of Hessenberg matrices which are rank-one perturbations of unitary matrices. The method is appealing because of its low memory requirements and low computational cost. In fact, the exploitation of the quasiseparable structure of the associated eigenvalue problems leads to a $O(n^2)$ complexity algorithm requiring only $O(n)$ memory space. The results of extensive numerical experiments confirm the robustness and effectiveness of the proposed approach. The accuracy of computed results is generally in accordance with the estimates on the conditioning of the input matrix.

**Acknowledgment.** The authors wish to thank the referees for their knowledgeable and helpful suggestions.

## REFERENCES

[1] G. S. AMMAR, D. CALVETTI, W. B. GRAGG, AND L. REICHEL, *Polynomial zerofinders based on Szegő polynomials*, J. Comput. Appl. Math., 127 (2001), pp. 1–161.
[2] G. S. AMMAR, D. CALVETTI, AND L. REICHEL, *Continuation methods for the computation of zeros of Szegő polynomials*, Linear Algebra Appl., 249 (1996), pp. 125–155.
[3] G. S. AMMAR, W. B. GRAGG, AND L. REICHEL, *Downdating of Szegő polynomials and data-fitting applications*, Linear Algebra Appl., 172 (1992), pp. 315–336.
[4] D. BINDEL, J. DEMMEL, W. KAHAN, AND O. MARQUES, *On computing Givens rotations reliably and efficiently*, ACM Trans. Math. Software, 28 (2002), pp. 206–238.
[5] D. A. BINI, F. DADDI, AND L. GEMIGNANI, *On the shifted QR iteration applied to companion matrices*, Electron. Trans. Numer. Anal., 18 (2004), pp. 137–152.

[6] D. Calvetti, S. Kim, and L. Reichel, *The restarted QR-algorithm for eigenvalue computation of structured matrices*, J. Comput. Appl. Math., 149 (2002), pp. 415–422.

[7] P. Dewilde and A.-J. van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.

[8] Y. Eidelman and I. Gohberg, *On a new class of structured matrices*, Integral Equations Operator Theory, 34 (1999), pp. 293–324.

[9] Y. Eidelman and I. Gohberg, *Linear complexity inversion algorithms for a class of structure matrices*, Integral Equations Operator Theory, 35 (1999), pp. 28–52.

[10] Y. Eidelman and I. Gohberg, *Fast inversion algorithms for a class block structured matrices*, Contemp. Math., 281 (2001), pp. 17–38.

[11] Y. Eidelman and I. Gohberg, *A modification of the Dewilde-van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl., 343+344 2002), pp. 419–450.

[12] P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, *Methods for modifying matrix factorizations*, Math. Comp., 28 (1974), pp. 505–535.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., John Hopkins University Press, 1996.

[14] W. B. Gragg, *The QR algorithm for unitary Hessenberg matrices*, J. Comput. Appl. Math., 16 (1986), pp. 1–8.

[15] W. B. Gragg, *Stabilization of the UHQR-algorithm*, in Advances in Computational Mathematics (Guangzhou, 1997), Lecture Notes in Pure and Appl. Math. 202, Dekker, New York, 1999, pp. 139–154.

[16] M. S. Moonen, G. H. Golub, and B. L. R. De Moor, eds., *Direct and inverse unitary eigenvalue problems in signal processing: an overview*, NATO Advanced Science Institutes Series E: Applied Sciences 232, Kluwer Academic Publishers, Dordrecht, 1993.

[17] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics 20, SIAM, 1998.

[18] L. Reichel, G. S. Ammar, and W. B. Gragg, *Discrete least squares approximation by trigonometric polynomials*, Math. Comp., 57 (1991), pp. 273–289.

[19] M. Stewart, *Stability properties of several variants of the unitary Hessenberg QR algorithm*, in Structured Matrices in Mathematics, Computer Science, and Engineering, II (Boulder, CO, 1999), Contemp. Math. 281, Amer. Math. Soc., Providence, RI, 2001, pp. 57–72.

[20] F. Tisseur, *Backward Stability of the QR Algorithm*, TR 239, UMR 5585, Universite de Saint Etienne, Lyon Saint-Etienne, France, 1996.

[21] E. E. Tyrtyshnikov, *Mosaic ranks for weakly semiseparable matrices*, in Large-Scale Scientific Computations of Engineering and Environmental Problems, II (Sozopol, 1999), Notes Numer. Fluid Mech. 73, Vieweg, Braunschweig, 2000, pp. 36–41.

[22] R. Vandebril, M. Van Barel, and N. Mastronardi, *A note on the representation and definition of semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 839–858.

[23] T.-L. Wang and W. B. Gragg, *Convergence of the shifted QR algorithm for unitary Hessenberg matrices*, Math. Comp., 71 (2003), pp. 1473–1496.

[24] T.-L. Wang and W. B. Gragg, *Convergence of the unitary QR algorithm with a unimodular Wilkinson shift*, Math. Comp., 72 (2003), pp. 375–385.

[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, in Monographs on Numerical Analysis, The Clarendon Press, Oxford University Press, New York, 1988.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 586–595

# CONVEXITY AND LIPSCHITZ BEHAVIOR OF SMALL PSEUDOSPECTRA*

J. V. BURKE†, A. S. LEWIS‡, AND M. L. OVERTON§

**Abstract.** The $\epsilon$-pseudospectrum of a matrix $A$ is the subset of the complex plane consisting of all eigenvalues of complex matrices within a distance $\epsilon$ of $A$, measured by the operator 2-norm. Given a nonderogatory matrix $A_0$, for small $\epsilon > 0$, we show that the $\epsilon$-pseudospectrum of any matrix $A$ near $A_0$ consists of compact convex neighborhoods of the eigenvalues of $A_0$. Furthermore, the dependence of each of these neighborhoods on $A$ is Lipschitz.

**Key words.** pseudospectrum, eigenvalue optimization, nonsmooth analysis, Lipschitz multifunction, robust optimization

**AMS subject classifications.** Primary, 15A18, 65K05; Secondary, 90C30, 93D09

**DOI.** 10.1137/050645841

## 1. Introduction.

Given a matrix $A$ in the space of $n \times n$ complex matrices $\mathbf{M}^n$, the spectrum $\Lambda(A)$ is an informative analytic tool, but must be interpreted with care. In particular, when $A$ has a multiple eigenvalue, small perturbations cause the spectrum to behave in a non-Lipschitz fashion.

Pseudospectra are robust analogs of the spectrum, enjoying many useful modelling properties. A comprehensive reference is [9]. We denote the operator 2-norm on $\mathbf{M}^n$ by $\|\cdot\|$. For real $\epsilon > 0$, the $\epsilon$ [illegible] of $A$ is the subset of the complex plane consisting of all eigenvalues of all complex matrices within a distance $\epsilon$ of $A$, measured by the operator 2-norm:

$$\Lambda_\epsilon(A) = \bigcup_{\|X-A\|\le\epsilon} \Lambda(X). \tag{1.1}$$

This subset of the complex plane $\mathbf{C}$ is semialgebraic (meaning that it can be described as a finite union of sets each defined via finitely many polynomial inequalities [2]) and consists of at most $n$ connected components; each component is compact and contains an eigenvalue of $A$. Visual plots of pseudospectra are richly informative and are conveniently computable via the EigTool package [5]. Note that, by contrast to our definition, [9] defines pseudospectra via the [illegible] inequality $\|X - A\| < \epsilon$.

Our aim in this work is to show how shifting attention from the spectrum to pseudospectra has a regularizing effect on variational behavior. Specifically, for matrices $A$ that are in a certain sense typical, even in the presence of multiple eigenvalues, if

---

*Received by the editors November 22, 2005, accepted for publication (in revised form) by A. J. Wathen October 17, 2006; published electronically May 1, 2007.
http://www.siam.org/journals/simax/29-2/64584.html

†Department of Mathematics, University of Washington, Seattle, WA 98195 (burke@math.washington.edu). This author's research was supported in part by National Science Foundation grant DMS-0505712.

‡School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14853 (aslewis@orie.cornell.edu, http://people.orie.cornell.edu~aslewis). This author's research was supported in part by National Science Foundation grant DMS-0504032.

§Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (overton@cs.nyu.edu). This author's research was supported in part by National Science Foundation grant DMS-0412049.

the parameter $\epsilon$ is small, the $\epsilon$-pseudospectrum consists of compact convex neighborhoods of the eigenvalues and varies in a Lipschitz fashion with respect to the Hausdorff distance.

## 2. Examples.

We begin with two examples to illustrate the potential difficulties. We first observe how the pseudospectrum can vary in a non-Lipschitz fashion even around a two-by-two matrix with simple eigenvalues. Second, we note that the component of the $\epsilon$-pseudospectrum containing a derogatory eigenvalue may fail to be convex, no matter how small the parameter $\epsilon > 0$.

To help our calculations, we use a well-known description of the pseudospectrum, more convenient than the definition (1.1). Denoting the smallest singular value by $\sigma_{\min} : \mathbf{M}^n \to \mathbf{R}$, the pseudospectrum is related to the reciprocal of the norm of the resolvent,

$$\sigma_{\min}(A - zI) = \|(A - zI)^{-1}\|^{-1},$$

via the useful characterization

$$\Lambda_\epsilon(A) = \{z \in \mathbf{C} : \sigma_{\min}(A - zI) \le \epsilon\}.$$

For our first example, we consider the behavior of the pseudospectrum $\Lambda_{\phi-1}(\cdot)$, where $\phi$ is the golden ratio $(1 + \sqrt{5})/2$, for matrices close to

$$\widehat{A} = \begin{bmatrix} 1 & 1 \\ 0 & -1 \end{bmatrix}.$$

An elementary calculation shows, for real $r$ and $\theta$, the formula

$$2\sigma_{\min}^2(\widehat{A} - re^{i\theta}I) \;=\; 3 + 2r^2 - \sqrt{5 + 4r^2(3 + 2\cos 2\theta)}. \tag{2.1}$$

This leads to a description of the pseudospectrum of $\widehat{A}$:

$$\Lambda_{\phi-1}(\widehat{A}) \;=\; \{re^{i\theta} : r^2 \le 2(2 - \phi + \cos 2\theta)\}.$$

The boundary of this set is a lemniscate centered at zero (see Figure 2.1); its interior consists of two disjoint open sets, each containing one of the eigenvalues $\pm 1$. In particular, notice that the pseudospectrum is contained in its tangent cone at zero:

$$\Lambda_{\phi-1}(\widehat{A}) \;\subset\; \{re^{i\theta} : \cos 2\theta \ge \phi - 2\}. \tag{2.2}$$

Now consider the point $ri$ on the imaginary axis as $r \downarrow 0$. The inclusion (2.2) implies a lower bound on the distance from this point to the pseudospectrum $\Lambda_{\phi-1}(\widehat{A})$ of the form

$$d(ri, \Lambda_{\phi-1}(\widehat{A})) \ge \alpha r, \tag{2.3}$$

for some constant $\alpha > 0$. On the other hand, formula (2.1) implies

$$\sigma_{\min}(\widehat{A} - re^{i\theta}I) = \phi - 1 + O(r^2), \tag{2.4}$$

so for some constant $\beta > 0$ we know that

$$ri \in \Lambda_{\phi-1+\beta r^2}(\widehat{A}).$$

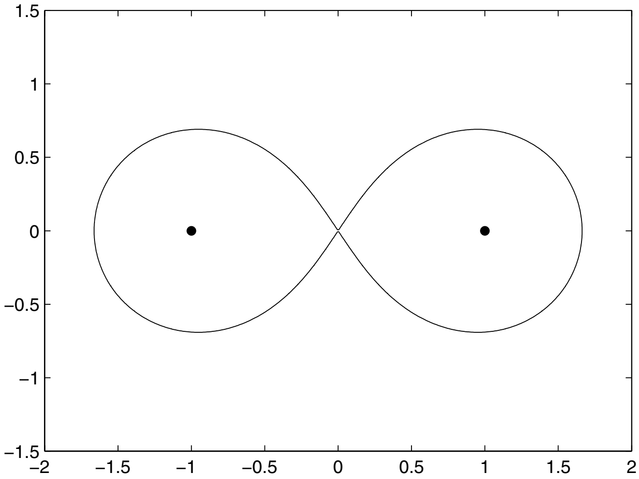


FIG. 2.1. *Pseudospectrum of $\hat{A}$.*

Using the definition of the pseudospectrum (1.1), we can rewrite the right-hand side as

$$\bigcup_{\|A-\widehat{A}\|\leq\beta r^2} \Lambda_{\phi-1}(A),$$

so there exists a matrix $A_r$ satisfying

$$\|A_r - \widehat{A}\| \leq \beta r^2 \quad \text{and} \quad ri \in \Lambda_{\phi-1}(A_r). \tag{2.5}$$

The [illegible] between two nonempty sets $K, L \subset \mathbf{C}$ is the quantity

$$H(K,L) = \max\left\{\sup_{z\in K} d(z,L), \sup_{z\in L} d(z,K)\right\},$$

where $d(z,L)$ is the distance from $z$ to $L$. Now, in conjunction with inequality (2.3), the relationships (2.5) imply that the Hausdorff distance between the pseudospectra $\Lambda_{\phi-1}(\widehat{A})$ and $\Lambda_{\phi-1}(A_r)$ is at least $\alpha r$, and yet the distance between the matrices $\widehat{A}$ and $A_r$ is at most $\beta r^2$. Thus the variation of the mapping $\Lambda_{\phi-1}$ around $\widehat{A}$ is not Lipschitz.

The pathology in this example is caused by the existence of a critical point of the function $z \mapsto \sigma_{\min}(\widehat{A} - zI)$ at a point on the boundary of the pseudospectrum (in this case $z = 0$): this can be seen directly from formula (2.4), or by observing that the left and right singular vectors of the matrix $\widehat{A}$ corresponding to the smallest singular value $\phi - 1$ are orthogonal (see [3, Cor. 7.2]). A direct calculation is also illuminating. Since $\sigma_{\min}(\widehat{A}) = \phi - 1$, replacing by zero the diagonal entry $\phi - 1$ in the singular value decomposition of $\widehat{A}$ makes a perturbation of size $\phi - 1$ and results in a singular matrix. But a straightforward calculation shows that this singular matrix is similar to a two-by-two Jordan block, so further perturbations of size $\delta$ result in the zero eigenvalue splitting into two distinct eigenvalues of size proportional to $\sqrt{\delta}$. It is this splitting that causes the pseudospectrum to behave in a non-Lipschitz fashion. In the development that follows, we avoid this possibility by focusing on the case of small $\epsilon$.

We discuss various aspects of the growth of pseudospectra as the parameter $\epsilon$ grows in a forthcoming work [4]. In particular, we can quantitatively estimate the component of the pseudospectrum $\Lambda_\epsilon(A)$ containing the eigenvalue $\lambda$: classical eigenvalue perturbation theory shows that the component approximates a disk of radius $(\alpha\epsilon)^{1/m}$ as $\epsilon \downarrow 0$, where $m$ is the multiplicity of $\lambda$ as a root of the minimal polynomial for $A$, and $\alpha$ is its associated condition number [4].

Despite approximating disks, small pseudospectral components may be nonconvex in general, as shown by our second example, suggested by [8]. Consider the matrix

$$\tilde{A} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

An easy calculation shows

$$\begin{aligned} f(r,\theta) &= \sigma^2_{\min}(\tilde{A} - re^{i\theta}I) = 1 - r\cos\theta + r^2 - \sqrt{(1 - r\cos\theta)^2 + r^2} \\ &\qquad\qquad\qquad\qquad = \frac{r^2}{2}(1 - r\cos\theta) + O(r^4) \ \text{ as } r \downarrow 0. \end{aligned}$$

Hence the component of the pseudospectrum $\Lambda_\epsilon(\tilde{A})$ containing zero, which we denote $\Lambda^0_\epsilon(\tilde{A})$, is a slightly distorted disk centered at zero and with radius approximately $\sqrt{2}\epsilon$, for small $\epsilon > 0$.

When $\theta = \pi/2$, another calculation shows

$$\frac{\partial f}{\partial r} = 2r\Big(1 - \frac{1}{\sqrt{1 + r^2}}\Big) > 0 \ \text{ for all } r > 0.$$

Hence for $\theta$ near $\pi/2$, the equation $f(r,\theta) = \epsilon^2$ implicitly defines $r$ as a smooth function $g(\theta)$, and for $r$ near $g(\pi/2) = \sqrt{2}\epsilon + O(\epsilon^2)$, the pseudospectrum is

$$\{re^{i\theta} : r \le g(\theta)\}.$$

One more calculation shows

$$g'(\pi/2) \ = \ \frac{\sqrt{1 + g^2(\pi/2)} - 1}{2\sqrt{1 + g^2(\pi/2)} - 1} \ = \ \epsilon^2 + O(\epsilon^3).$$

To summarize, the pseudospectral boundary for the matrix $\tilde{A}$ crosses the positive imaginary axis at a unique point $z_\epsilon = (\sqrt{2}\epsilon + O(\epsilon^2))i$. The boundary nearby is a smooth curve crossing the imaginary axis nonorthogonally and bounding the pseudospectral component below it. Clearly, exactly the same properties hold for the matrix $-\tilde{A}$, and the two boundaries are mirror images in the imaginary axis. Finally, consider the matrix

$$A = \begin{bmatrix} \tilde{A} & 0 \\ 0 & -\tilde{A} \end{bmatrix}.$$

Since the singular values of block-diagonal matrices are just the singular values of the blocks, we have $\Lambda_\epsilon(A) = \Lambda_\epsilon(\tilde{A}) \cup \Lambda_\epsilon(-\tilde{A})$, so we know $\Lambda^0_\epsilon(A) = \Lambda^0_\epsilon(\tilde{A}) \cup \Lambda^0_\epsilon(-\tilde{A})$. By considering a neighborhood of the point $z_\epsilon$, this latter set cannot be convex.

In this example the difficulty is caused by the fact that the zero eigenvalue is derogatory. In what follows, we show good behavior of pseudospectra around nonderogatory eigenvalues, providing the parameter $\epsilon$ is sufficiently small.

**3. Background results.** We recall some results from [3]. A real-valued function on a real vector space is [illegible] at zero if in some neighborhood of zero it can be written as the sum of an absolutely convergent power series in the coordinates relative to some basis, and we make an analogous definition at other points. In particular, such functions are $C^\infty$ near the point in question.

The smallest singular value of the matrix $Z$ is [illegible] when the smallest eigenvalue of the Hermitian matrix $Z^*Z$ is simple. Since the eigenvalues of matrices depend continuously on the matrix, the set of matrices $Z$ with simple smallest singular values is open.

We consider the function $h : \mathbf{M}^n \times \mathbf{C} \to \mathbf{R}$ defined by

$$h(A, z) = (\sigma_{\min}(A - zI))^2.$$

For any $A \in \mathbf{M}^n$, we also define a function $h_A : \mathbf{C} \to \mathbf{R}$ by $h_A(z) = h(A, z)$. Treating $\mathbf{C}$ as a Euclidean space with inner product $\langle w, z\rangle = \operatorname{Re}(w^*z)$, we can interpret gradients $\nabla h_A(z)$ as elements of $\mathbf{C}$.

Theorem 3.1 (analytic singular value). [illegible] $Z$ [illegible] $\sigma^2_{\min}$ [illegible] $Z$

An eigenvalue of $A$ is [illegible] if it has geometric multiplicity one. Among multiple eigenvalues, the nonderogatory ones are the most typical (from the perspective of the dimensions of the corresponding manifolds in $\mathbf{M}^n$ [1]). The matrix $A$ is [illegible] if all its eigenvalues are nonderogatory.

The following result is very well known.

Proposition 3.2 (nonderogatory eigenvalues). [illegible] $A$ [illegible] $\lambda$ [illegible] $A - \lambda I$

The next result, an immediate consequence of [3, Thm. 7.4 and Cor. 7.8], shows that the resolvent norm is well-behaved near any nonderogatory eigenvalue of $A$. For a symmetric matrix $X$, we write $X \succ 0$ to mean $X$ is positive-definite.

Theorem 3.3 (growth near an eigenvalue). [illegible] $\lambda$ [illegible] $A$ [illegible] $z \neq \lambda$ [illegible] $\lambda$ [illegible] $h_A$ [illegible] $\nabla h_A(z) \neq 0$ [illegible] $\nabla^2 h_A(z) \succ 0$

Related results appear in [6].

**4. Convexity.** In [3] we observe, as a consequence of Theorem 3.3 (growth near an eigenvalue), that if $\lambda$ is a nonderogatory eigenvalue of a matrix $A$, then for small $\epsilon > 0$ the part of the pseudospectrum $\Lambda_\epsilon(A)$ near $\lambda$ is strictly convex. (We call a closed set $S \subset \mathbf{C}$ [illegible] if the open line segment $(u, v)$ lies in $\operatorname{int} S$ for any distinct points $u, v \in S$.) The first step in our development is to generalize this result to allow the matrix $A$ to vary. We denote the closed unit disk in $\mathbf{C}$ by $D$ and the closed unit ball in $\mathbf{M}^n$ by $B$.

We begin with a rather technical statement of our basic tool.

Theorem 4.1 (small pseudospectra). [illegible] $\lambda$ [illegible] $A_0 \in \mathbf{M}^n$ [illegible] $\mu > 0$ [illegible] $\bar\epsilon \in (0, \mu)$ [illegible] $\mu$ [illegible] $\epsilon \in (0, \bar\epsilon)$ [illegible]

1. [illegible] $A \in \mathbf{M}^n$ [illegible] $A_0$ [illegible] $\mu$ [illegible] $\epsilon$ [illegible]

$$\hat\Lambda_\epsilon(A) = \Big\{ z \in \Lambda_\epsilon(A) : |z - \lambda| < \mu \Big\} \tag{4.1}$$

[illegible] $\Lambda_\epsilon(A)$ [illegible] $\lambda$ [illegible] $A_0$ [illegible] $\lambda$

2. [illegible] $\bar{\eta} \in (0, \mu)$ [illegible] $\mu$ [illegible] $\epsilon$ [illegible] $\eta \in (0, \bar{\eta})$ [illegible] $A$ [illegible] $A_0$ [illegible] $\mu$ $\epsilon$ [illegible] $\eta$ [illegible]

(i) $\hat{\Lambda}_\epsilon(A)$ [illegible] $\lambda + \eta D$ [illegible] $z \in \lambda + \mu D$

(ii) [illegible] $A - zI$ [illegible]

(iii) [illegible] $|z - \lambda| \geq \eta$ [illegible] $\nabla h_A(z) \neq 0$ [illegible] $\nabla^2 h_A(z) \succ 0$

[illegible] Without loss of generality, $\lambda = 0$. By Theorem 3.3 (growth near an eigenvalue), there exists a number $\mu > 0$ such that

$$0 < |z| \leq \mu \;\; \Rightarrow \;\; \nabla h_{A_0}(z) \neq 0 \text{ and } \nabla^2 h_{A_0}(z) \succ 0. \tag{4.2}$$

Hence the function $h_{A_0}$ is strictly convex on the disk $\mu D$, with a strict local minimum value of zero at zero. In particular, we deduce

$$\Lambda(A_0) \cap \mu D = \{0\}. \tag{4.3}$$

Consider the open set

$$\Omega \;=\; \Big\{(A, z) \in \mathbf{M}^n \times \mathbf{C} : \text{the smallest singular value of } A - zI \text{ is simple}\Big\}.$$

Theorem 3.1 (analytic singular value) implies that the function $h$ is real-analytic throughout $\Omega$, so the function $(A, z) \mapsto \nabla^2 h_A(z)$ is continuous on $\Omega$. Clearly $(A_0, 0) \in \Omega$. Hence, by reducing $\mu$ if necessary, we can suppose there exists a number $\delta_1 > 0$ such that

$$\Big\{(A, z) \in \mathbf{M}^n \times \mu D : \|A - A_0\| < \delta_1\Big\} \;\subset\; \Omega.$$

Choose any number $\mu_1 \in (0, \mu)$. Then we claim

$$\Lambda_\epsilon(A_0) \subset \mu_1 D \cup \mu D^c \tag{4.4}$$

for all small $\epsilon > 0$ (where $D^c$ denotes the complement of $D$). If this were not the case, there would exist sequences of parameters $\epsilon_r \downarrow 0$ and points $z_r \in \Lambda_{\epsilon_r}(A_0)$ satisfying $\mu_1 < |z_r| \leq \mu$. By compactness, we can suppose $z_r$ approaches a nonzero point $z \in \mu D$. However, since $\sigma_{\min}(A_0 - z_r I) \leq \epsilon_r$ for all $r$, we then deduce $\sigma_{\min}(A_0 - zI) \leq 0$, so $z \in \Lambda(A_0)$, contradicting (4.3).

Fix any $\epsilon > 0$ small enough to ensure inclusion (4.4), and choose any number $\mu_2 \in (\mu_1, \mu)$. We claim there exists a number $\delta_2 \in (0, \delta_1)$ such that

$$\|A - A_0\| < \delta_2 \;\Rightarrow\; \hat{\Lambda}_\epsilon(A) \subset \mu_2 D. \tag{4.5}$$

Indeed, if this fails, there are sequences of matrices $A_r \to A_0$ and points $z_r \in \hat{\Lambda}_\epsilon(A_r)$ satisfying $\mu_2 < |z_r| < \mu$. By compactness, we can suppose $z_r$ approaches a point $z \in \mu D$ satisfying $|z| \geq \mu_2 > \mu_1$. However, since $\sigma_{\min}(A_r - z_r I) \leq \epsilon$ for all $r$, we deduce $\sigma_{\min}(A_0 - zI) \leq \epsilon$, and hence $z \in \Lambda_\epsilon(A_0)$. But this contradicts inclusion (4.4).

The inclusion $\hat{\Lambda}_\epsilon(A) \subset \mu_2 D$ implies that the set $\hat{\Lambda}_\epsilon(A)$ is compact, being the intersection of the two compact sets $\Lambda_\epsilon(A)$ and $\mu_2 D$.

For our next step, observe that, by continuity, we know there exists a number $\eta \in (0, \mu)$ such that $\sigma_{\min}(A_0 - zI) < \epsilon$ for all points $z \in \eta D$. We now claim there exists a number $\delta_3 \in (0, \delta_2)$ such that

$$\|A - A_0\| < \delta_3 \;\Rightarrow\; \eta D \subset \operatorname{int} \hat{\Lambda}_\epsilon(A). \tag{4.6}$$

Suppose this property fails, so there are sequences of matrices $A_r \to A_0$ and points $z_r \in \eta D$ satisfying $z_r \notin \operatorname{int} \hat{\Lambda}_\epsilon(A_r)$, and hence $\sigma_{\min}(A_r - z_r I) \geq \epsilon$. By compactness, we can suppose $z_r$ approaches a point $z \in \eta D$, giving the contradiction $\sigma_{\min}(A_0 - zI) \geq \epsilon$.

We next claim there exists a number $\delta \in (0, \delta_3)$ such that, whenever $\|A - A_0\| \leq \delta$ and $\eta \leq |z| \leq \mu$, we have

$$\nabla h_A(z) \neq 0 \quad \text{and} \quad \nabla^2 h_A(z) \succ 0. \tag{4.7}$$

If this fails, there are sequences of matrices $A_r \to A_0$ and points $z_r$ satisfying $\eta \leq |z_r| \leq \mu$ and

$$\min \Big\{ |\nabla h_{A_r}(z_r)|, \lambda_{\min}(\nabla^2 h_{A_r}(z_r)) \Big\} \leq 0$$

for all $r$. By compactness, we can suppose $z_r$ approaches a point $\hat{z}$ satisfying $\eta \leq |\hat{z}| \leq \mu$. By the continuity with respect to $(A, z) \in \Omega$ of the functions $\nabla h_A(z)$ and $\nabla^2 h_A(z)$, we deduce

$$\min \Big\{ |\nabla h_{A_0}(\hat{z})|, \lambda_{\min}(\nabla^2 h_{A_0}(\hat{z})) \Big\} \leq 0,$$

contradicting statement (4.2).

We next prove that the set $\hat{\Lambda}_\epsilon(A)$ is strictly convex. To this end, consider any matrix $A$ satisfying $\|A - A_0\| < \delta$ and any two distinct points $u, v \in \hat{\Lambda}_\epsilon(A)$. We want to show the open line segment $(u, v)$ lies in $\operatorname{int} \hat{\Lambda}_\epsilon(A)$. By property (4.6), we know that

$$\eta D \subset \operatorname{int} \hat{\Lambda}_\epsilon(A). \tag{4.8}$$

We consider various cases.

(i) $|u|, |v| \leq \eta$. The result then follows by inclusion (4.8).

(ii) $(u, v) \cap \eta D = \emptyset$. In this case, we know $h_A(u) \leq \epsilon^2$ and $h_A(v) \leq \epsilon^2$, and the function $h_A$ is strictly convex on the line segment $[u, v]$, by property (4.7), so the result follows.

(iii) $|u| \leq \eta$ and $|v| > \eta$. Then consider the unique number $\gamma \in [0, 1]$ such that the point $w = \gamma u + (1 - \gamma)v$ satisfies $|w| = \eta$. Then $[u, w] \subset \operatorname{int} \hat{\Lambda}_\epsilon(A)$ by inclusion (4.8), while $(w, v) \subset \operatorname{int} \hat{\Lambda}_\epsilon(A)$ by case (ii).

(iv) $|u| > \eta$ and $|v| \leq \eta$. By swapping $u$ and $v$, we obtain case (iii).

(v) $|u|, |v| > \eta$ and $(u, v) \cap \eta D \neq \emptyset$. Consider the two (possibly equal) solutions $\gamma_1 \geq \gamma_2$ in $[0, 1]$ to the quadratic equation $|\gamma u + (1 - \gamma)v|^2 = \eta^2$. For each $j = 1, 2$, set $w_j = \gamma_j u + (1 - \gamma_j)v$. Then $[w_1, w_2] \subset \operatorname{int} \hat{\Lambda}_\epsilon(A)$ by inclusion (4.8), while both intervals $(u, w_1)$ and $(w_2, v)$ lie in $\operatorname{int} \hat{\Lambda}_\epsilon(A)$ by case (ii).

This completes the proof of strict convexity.

To see that the set $\hat{\Lambda}_\epsilon(A)$ must be the component of the pseudospectrum $\Lambda_\epsilon(A)$ containing $\lambda$, note that the function $A \mapsto \sigma_{\min}(A - \lambda I)$ is continuous on $\mathbf{M}^n$, and $\sigma_{\min}(A_0 - \lambda I) = 0$, so $\lambda \in \hat{\Lambda}_\epsilon(A)$ for all $A$ near $A_0$. Since the $\hat{\Lambda}_\epsilon(A)$ is a connected subset of $\Lambda_\epsilon(A)$, being convex, the result follows. $\square$

COROLLARY 4.2 (strict convexity). [illegible] $\lambda$ [illegible] $A_0 \in \mathbf{M}^n$ [illegible] $\epsilon > 0$ [illegible] $\Lambda_\epsilon(A)$ [illegible] $\lambda$ [illegible] $A$ [illegible] $A_0$

**5. Sensitivity.** We are now ready to study the dependence of a fixed component of the pseudospectrum $\Lambda_\epsilon(A)$ on the matrix $A$.

LEMMA 5.1 (gradient continuity). [illegible] 4.1 [illegible] $w$ [illegible] $\alpha_w : \mathbf{M}^n \to \mathbf{R}$ [illegible]

$$\alpha_w(A) = \sup\{\mathrm{Re}\,(w^* z) : z \in \hat{\Lambda}_\epsilon(A)\}. \tag{5.1}$$

[illegible] $(A, w) \mapsto \alpha_w(A)$ [illegible] $C^\infty$ [illegible]

$$\Big\{(A, w) \in \mathbf{M}^n \times \mathbf{C} : \|A - A_0\| \le \delta,\ w \ne 0\Big\}.$$

[illegible]. The supremum (5.1) is attained at a unique point $z(A, w) \in \hat{\Lambda}_\epsilon(A)$, since the set $\hat{\Lambda}_\epsilon(A)$ is compact and strictly convex. We can also write the supremum as a smooth optimization problem,

$$\alpha_w(A) = \sup\Big\{\mathrm{Re}\,(w^* z) : h_A(z) \le \epsilon^2,\ |z - \lambda| < \mu\Big\}.$$

By continuity, the optimal solution $z(A, w)$ must satisfy $h_A(z(A, w)) = \epsilon^2$. The function $h_A$ is real-analytic (so, in particular, $C^\infty$), and satisfies the condition

$$\nabla h_A(z(A, w)) \ne 0 \ \text{ and } \ \nabla^2 h_A(z(A, w)) \succ 0. \tag{5.2}$$

We now apply a standard sensitivity argument to show that the dependence of the optimal solution $z(A, w)$ on the parameters $(A, w)$ is also $C^\infty$. We argue as follows. Since $\nabla h_A(z(A, w)) \ne 0$, there exists a Lagrange multiplier $\gamma(A, w) \in \mathbf{R}$ corresponding to the optimal solution. Thus $z = z(A, w)$ and $\gamma = \gamma(A, w)$ solve the system

$$\begin{aligned} w + \gamma \nabla h_A(z) &= 0, \\ h_A(z) &= \epsilon^2. \end{aligned}$$

But it is easy to check that condition (5.2) implies that the Jacobian for the left-hand side is surjective at $(z(A, w), \gamma(A, w))$. Hence the implicit function theorem implies that the mapping $(A, w) \mapsto z(A, w)$ is $C^\infty$. The result follows. ◻

We can now prove our main result.

THEOREM 5.2 (component Lipschitz behavior). [illegible] $\lambda$ [illegible] $A_0 \in \mathbf{M}^n$ [illegible] $\mu > 0$ [illegible] $\bar{\epsilon} \in (0, \mu)$ [illegible] $\mu$ [illegible] $\epsilon \in (0, \bar{\epsilon})$ [illegible] $A \in \mathbf{M}^n$ [illegible] $A_0$ [illegible] $\mu$ [illegible] $\epsilon$ [illegible]

$$\hat{\Lambda}_\epsilon(A) \ = \ \Big\{z \in \Lambda_\epsilon(A) : |z - \lambda| < \mu\Big\}$$

[illegible]

(i) $\hat{\Lambda}_\epsilon(A)$ [illegible] $\Lambda_\epsilon(A)$ [illegible] $\lambda$

(ii) $\hat{\Lambda}_\epsilon(A)$ [illegible] $A_0$ [illegible] $\lambda$

(iii) $\hat{\Lambda}_\epsilon(A)$ [illegible]

(iv) [illegible] $\hat{\Lambda}_\epsilon$ [illegible] $A_0$ [illegible]

[illegible]. We apply Theorem 4.1 (small pseudospectra) and Corollary 4.2 (strict convexity). Using Lemma 5.1 (gradient continuity), we can define a number

$$L = \max\Big\{\|\nabla\alpha_w(A)\| : A \in \Gamma,\ |w| = 1\Big\},$$

where the set $\Gamma$ is the neighborhood of the matrix $A_0$ referred to in Theorem 4.1. Consider any two matrices $A_1, A_2 \in \Gamma$. According to [7, Lemma 2], the Hausdorff distance between the sets $\hat{\Lambda}_\epsilon(A_1)$ and $\hat{\Lambda}_\epsilon(A_2)$ is given by

$$\max_{|w|=1} \Big|\alpha_w(A_1) - \alpha_w(A_2)\Big|,$$

and, by the definition of $L$, this quantity cannot exceed $L\|A_1 - A_2\|$. ◻

In particular, we obtain the following variational property of the entire pseudopectrum.

COROLLARY 5.3 (pseudospectral Lipschitz behavior). [illegible] $A_0 \in \mathbf{M}^n$ [illegible] $\epsilon > 0$ [illegible] $\Lambda_\epsilon(A)$ [illegible] $A \in \mathbf{M}^n$ [illegible] $A_0$ [illegible]

[illegible]. Denote the distinct eigenvalues of $A_0$ by $\lambda_1, \lambda_2, \ldots, \lambda_m$, and denote the separation of the eigenvalues by $\nu = \min_{j\neq k} |\lambda_j - \lambda_k|$. Now apply the preceding result successively at each eigenvalue $\lambda_j$ to obtain a number $\mu < \nu/3$ such that any small $\epsilon > 0$ has the following property: for all matrices $A$ near $A_0$ and each index $j = 1, 2, \ldots, m$, the component of the pseudospectrum $\Lambda_\epsilon(A)$ containing $\lambda_j$ is

$$\Lambda_\epsilon^j(A) = \{z \in \Lambda_\epsilon(A) : |z - \lambda_j| < \mu\},$$

and the set-valued mapping $\Lambda_\epsilon^j$ is Lipschitz around $A_0$.

Now consider any matrices $A_1, A_2 \in \mathbf{M}^n$ near $A_0$. For any fixed index $j$, we have

$$z \in \Lambda_\epsilon^j(A_1) \quad \Rightarrow \quad d(z, \Lambda_\epsilon(A_2)) = d(z, \Lambda_\epsilon^j(A_2)). \tag{5.3}$$

To see this, notice that $d(z, \Lambda_\epsilon^j(A_2)) < \mu$ because $\lambda_j \in \Lambda_\epsilon^j(A_2)$. On the other hand, for indices $k \neq j$, we know

$$|z - \lambda_j| < \mu, \quad |\lambda_j - \lambda_k| > 3\mu, \quad \Lambda_\epsilon^k(A_2) \subset \lambda_k + \mu D,$$

so $d(z, \Lambda_\epsilon^k(A_2)) > \mu$. Since

$$d(z, \Lambda_\epsilon(A_2)) = \min_k d(z, \Lambda_\epsilon^k(A_2)),$$

our claim (5.3) now follows.

As a consequence of the implication (5.3), we obtain

$$\sup_{z\in\Lambda_\epsilon(A_1)} d(z, \Lambda_\epsilon(A_2)) = \max_j \sup_{z\in\Lambda_\epsilon^j(A_1)} d(z, \Lambda_\epsilon(A_2)) = \max_j \sup_{z\in\Lambda_\epsilon^j(A_1)} d(z, \Lambda_\epsilon^j(A_2)),$$

and similarly,

$$\sup_{z\in\Lambda_\epsilon(A_2)} d(z, \Lambda_\epsilon(A_1)) = \max_k \sup_{z\in\Lambda_\epsilon^k(A_2)} d(z, \Lambda_\epsilon^k(A_1)).$$

Hence the Hausdorff distance between the pseudospectra $\Lambda_\epsilon(A_1)$ and $\Lambda_\epsilon(A_2)$ is given by

$$\begin{aligned} H\Big(\Lambda_\epsilon(A_1), \Lambda_\epsilon(A_2)\Big) &= \max\Big\{ \sup_{z\in\Lambda_\epsilon(A_1)} d(z, \Lambda_\epsilon(A_2)), \sup_{z\in\Lambda_\epsilon(A_2)} d(z, \Lambda_\epsilon(A_1))\Big\} \\ &= \max\Big\{ \max_j \sup_{z\in\Lambda_\epsilon^j(A_1)} d(z, \Lambda_\epsilon^j(A_2)), \max_k \sup_{z\in\Lambda_\epsilon^k(A_2)} d(z, \Lambda_\epsilon^k(A_1))\Big\} \\ &= \max_r \ \max\Big\{ \sup_{z\in\Lambda_\epsilon^r(A_1)} d(z, \Lambda_\epsilon^r(A_2)), \sup_{z\in\Lambda_\epsilon^r(A_2)} d(z, \Lambda_\epsilon^r(A_1))\Big\} \\ &= \max_r H\Big(\Lambda_\epsilon^r(A_1), \Lambda_\epsilon^r(A_2)\Big). \end{aligned}$$

The result now follows. □

**Acknowledgments.** The authors thank Mark Embree and Nick Trefethen for a number of insights that improved the overall presentation.

**Note added in proof.** A generalization of Corollary 5.3 to the derogatory case appears in a recent preprint, "Variational Analysis of Pseudospectra" by A. S. Lewis and C. H. J. Pang.

## REFERENCES

[1] V. I. Arnold, *On matrices depending on parameters*, Uspehi Mat. Nauk, 26 (1971), pp. 29–43.
[2] R. Benedetti and J.-J. Risler, *Real Algebraic and Semi-Algebraic Sets*, Hermann, Paris, 1990.
[3] J. V. Burke, A. S. Lewis, and M. L. Overton, *Optimization and pseudospectra, with applications to robust stability*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 80–104. Corrigendum: www.cs.nyu.edu/cs/faculty/overton/papers/pseudo_corrigendum.html.
[4] J. V. Burke, A. S. Lewis, and M. L. Overton, *Spectral conditioning and pseudospectral growth*, Numer. Math., to appear.
[5] M. Embree and L. N. Trefethen, *Pseudospectra Gateway*, web.comlab.ox.ac.uk/pseudospectra.
[6] M. Karow, *Eigenvalue condition numbers and a formula of Burke, Lewis and Overton*, Electron. J. Linear Algebra, 15 (2006), pp. 143–153.
[7] G. Salinetti and R. J.-B. Wets, *On the convergence of sequences of convex sets in finite dimensions*, SIAM Rev., 21 (1979), pp. 18–33.
[8] L. N. Trefethen, *private communication*, 2005.
[9] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, Princeton, NJ, 2005.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 596–605

# OPTIMAL EMBEDDINGS AND EIGENVALUES IN SUPPORT THEORY*

ERIK G. BOMAN[†], STEPHEN GUATTERY[‡], AND BRUCE HENDRICKSON[†]

**Abstract.** Support theory is a methodology for bounding eigenvalues and generalized eigenvalues of matrices and matrix pencils; such bounds have been stated both in algebraic terms and in combinatorial terms based on embeddings of the underlying graphs of the matrices. In this paper, we present a theorem that demonstrates the connection between these various bounding techniques and also suggest a possible approach to generating approximate inverses for preconditioning. The theorem shows, given matrices $A = UD_AU^*$ and $B = VD_BV^*$ (where $D_A$ and $D_B$ are invertible Hermitian matrices, and $U$ and $V$ are not necessarily square), that it is possible to define a matrix $W$ such that $W^*D_B^{-1}WD_A$ has the same nonzero eigenvalues counting multiplicity as $B^+A$. In the special case that $U$ is the orthogonal projector onto the range space of $B$ and $D_A = I$ (and hence that $A = UU^* = U^2 = U$), then $W^*D_B^{-1}W = B^+$. This suggests that finding an approximation to $W$ might lead to an approximate inverse that can be used in preconditioning. We also describe how this theorem generalizes the idea of graph embeddings in an algebraic sense.

**Key words.** Laplacian eigenvalues, generalized eigenvalues, support theory

**AMS subject classifications.** 05C50, 15A09, 15A18, 15A22, 65F10

**DOI.** 10.1137/050642174

## 1. Introduction.

Support theory is a methodology for bounding eigenvalues and generalized eigenvalues (and more generally support numbers) of matrices and matrix pencils; it has applications such as the analysis of the performance of interative solvers for symmetric positive definite systems. Support theory bounds have been stated both in algebraic terms and in terms of combinatorial techniques based on the underlying graphs of the matrices involved. In this paper, we present a theorem that demonstrates the connection between these various bounding techniques and also suggest a possible approach to generating approximate inverses for preconditioning.

Given a preconditioned system $B^{-1}A$, support theory is concerned with generating bounds on $\lambda_{\max}(B^{-1}A)$ and $\lambda_{\max}(A^{-1}B)$, which are used to bound the condition number of the system. More generally, if $A$ and $B$ are singular, bounds are generated on the support numbers $\sigma(A, B)$ and $\sigma(B, A)$. Bounds are generated in terms of factorizations $A = UU^T$ and $B = VV^T$ of the matrices where $U$ and $V$ are not restricted to be square. Given the factorizations, a matrix $W$ is constructed subject to the condition $U = VW$. This is the key formulation of support theory [3].

We present a theorem that shows, given matrices $A = UD_AU^*$ and $B = VD_BV^*$ (where $D_A$ and $D_B$ are invertible Hermitian matrices and the columns of $U$ are in the range of $B$), that there exists a matrix $W_{opt}$ such that $W_{opt}^*D_B^{-1}W_{opt}D_A$ has the same nonzero eigenvalues counting multiplicity as $B^+A$. This result was previously

---

*Received by the editors October 7, 2005; accepted for publication (in revised form) by I. S. Dhilion November 27, 2006; published electronically May 1, 2007.

http://www.siam.org/journals/simax/29-2/64217.html

[†]Discrete Algorithms and Math Department, Sandia National Laboratories, Albuquerque, NM 87185-1111 (egboman@sandia.gov, bahendr@sandia.gov). Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

[‡]Department of Computer Science, Bucknell University, Lewisburg, PA 17837 (guattery@bucknell.edu). Parts of this work were carried out while the author was on a sabbatical appointment with the Computer Science Research Institute, Sandia National Labs, New Mexico.

unknown even in the special case where $D_A = D_B = I$. In the special case that $U$ is the orthogonal projector of the range space of $B$ and $D_A = I$ (and hence that $A = UU^* = U^2 = U$), then $W_{opt}^* D_B^{-1} W_{opt} = B^+$, where $B^+$ is the pseudoinverse of $B$. This suggests that finding an approximation to $W_{opt}$ might lead to an approximate inverse that can be used in preconditioning.

We describe how this theorem generalizes the idea of embedding in an algebraic sense and show how it can be used to generalize and simplify the proofs of previous results. In particular, this allows the ideas to be applied to a broader range of matrices. We also place the theorem in the context of two branches of support theory research that focus on matrix eigenvalue bounds and matrix pencil bounds, respectively.

The paper starts with a section on notation (section 2) and some background on the development of some basic ideas in support theory (section 3). The main theorem is presented in section 4, along with a version applicable to non-Hermitian matrices. We use this proof to give a new and more general proof of a result by Boman and Hendrickson [3]. In section 6 we give new and more general proofs of results by Guattery [6] linking a generalized notion of embedding with the pseudoinverses of Hermitian matrices. The results in this paper subsume the results in that technical report. Finally, in section 7 we discuss the implications of this work for preconditioning, particularly in terms of approximate inverses.

## 2. Notation.

We use capital letters to represent matrices. Individual matrix entries are denoted by the corresponding lower case letter with subscripts showing the row and column of the entry, e.g., $a_{ij}$ is the entry of $A$ in row $i$ and column $j$. $I$ represents an identity matrix. When it is useful to indicate the size of an identity matrix, a single subscript indicates the number of rows and columns: $I_k$ is the $k \times k$ identity matrix. A matrix of all zeros is denoted by 0.

For a matrix $A$ with real entries, $|A|$ denotes the matrix whose entries are the absolute values of the corresponding entries of $A$: the entry in row $i$ and column $j$ of $|A|$ is $|a_{ij}|$.

The notations $A^T$ represents the transpose of $A$; $A^*$ represents $A$'s conjugate transpose; and $A^+$ represents the pseudoinverse of $A$ (i.e., the Moore–Penrose generalized inverse of $A$).

We denote the range space of the columns of a matrix $A$ by $R(A)$. The orthogonal projector onto a vector space $\mathcal{S}$ is denoted as $P_{\mathcal{S}}$. Thus $P_{R(A)}$ is the orthogonal projector onto the range space of the columns of $A$.

Vectors are denoted by lower case letters with an arrow above, e.g., $\vec{v}$. A column vector of all zeros is denoted as $\vec{0}$; if a specific size is specified, it is given as a subscript: $\vec{0}_k$ is a vector of $k$ 0's.

## 3. Background.

A key application of support theory is the analysis of preconditioned symmetric and Hermitian systems. When solving linear systems $Ax = b$ using an iterative method, it is frequently useful to have a good preconditioner to accelerate convergence. This has often involved constructing a preconditioner $B \approx A$. In this sense, $B$ is a good preconditioner if both (i) the eigenvalues of $B^{-1}A$ are clustered around one, and (ii) the matrix $B$ is easy to solve for (invert). (Note that if $A$ is singular, one may wish to let $B$ be singular with the same null space. In this case, $B^+A$ is the preconditioned matrix of interest.)

For Hermitian positive definite systems, the eigenvalues are real and positive, and a lower bound on the rate of convergence is the spectral condition number, $\kappa(C) = \lambda_{\max}(C)/\lambda_{\min}(C)$. In our case, either $C = B^{-1}A$ or $C = B^{-1/2}AB^{-1/2}$ (where $B^{-1/2}$ is Hermitian), and the condition number can be expressed using support numbers.

The [illegible] for a matrix pencil $(A, B)$, where $A$ and $B$ are Hermitian, is defined as

$$\sigma(A, B) = \min \{ t \in \mathbb{R} \,|\, x^*(\tau B - A)x \geq 0 \text{ for all } x \in \mathbb{C}^n \text{ and for all } \tau \geq t \} .$$

It has been shown that when $B$ is symmetric positive definite, $\sigma(A, B) = \lambda_{\max}(A, B)$, the largest generalized eigenvalue [3]. Furthermore,

$$\kappa(B^{-1/2}AB^{-1/2}) = \sigma(A, B)\sigma(B, A).$$

Support numbers exist even when $A$ or $B$ is singular, but may not be finite. Support theory is useful for analyzing preconditioners because support numbers give bounds on the spectral condition number. See [3] for further information on support theory.

For symmetric positive semidefinite systems, bounds on support numbers are generated in terms of factorizations $A = UU^T$ and $B = VV^T$ of the matrices. Based on the factorizations, a matrix $W$ is constructed subject to the condition $U = VW$. This key algebraic formulation of support theory is due to Boman and Hendrickson [3] and is at the heart of the symmetric product support theorem.

THEOREM 3.1 (see Theorem 4.5 from [3]). [illegible] $U \in \mathbb{R}^{n \times k}$ [illegible] $V \in \mathbb{R}^{n \times p}$ [illegible]

$$\sigma(UU^T, VV^T) = \min_W \|W\|_2^2 \quad \textit{subject to}\, VW = U.$$

It is further noted in [3] that for $\hat{W} = V^+U$, $\sigma(UU^T, VV^T) = \|\hat{W}\|_2^2$.

Boman and Hendrickson also proved the following theorem.

THEOREM 3.2 (see Theorem 4.7 from [3]). [illegible] $U \in \mathbb{R}^{n \times k}$ [illegible] $V \in \mathbb{R}^{n \times p}$ [illegible] $D \in \mathbb{R}^{k \times k}$ [illegible] $W$ [illegible] $VW = U$

$$\sigma(UDU^T, VV^T) \leq \lambda_{\max}(WDW^T) \leq \lambda_{\max}(D)\|W\|_2^2.$$

The use of the term [illegible] in this paper is intended to suggest the relationship to the use of combinatorial embeddings in early work in the area, which typically involved Laplacian (and related) matrices. Laplacians have a well-known graphical interpretation in which a Laplacian $L$ can be factored into $L = UU^T$, where $U$ is a [illegible]. In such a matrix, each row represents a vertex of the underlying graph of the Laplacian. Each column represents an edge and has one $+1$ and one $-1$ in the rows corresponding to the endpoints of the edge. Note that this (arbitrarily) directs the edges, though this direction is lost in the product that forms the Laplacian. (Entries can be scaled appropriately for weighted graphs.)

Graph embedding techniques were used to analyze the quality of support graph preconditioners, which were formulated in terms of the combinatorial structure of the Laplacian matrices. Examples include Vaidya's preconditioners based on spanning trees [13] (see [2] for a more readily available description of Vaidya's preconditioner) and Gremban and Miller's support tree preconditioners [5]. Bounds on the condition number of a preconditioned system were calculated in terms of properties of the embeddings of the underlying graphs of the preconditioner into the Laplacian and vice versa.

The embeddings used in this analysis were typically path embeddings, in which each edge in $U$ (e.g., the vertex-edge incidence matrix of the original Laplacian) was represented as a directed path constructed from the edges in $V$ (e.g., the vertex-edge incidence matrix of the preconditioner). While these embeddings were not typically expressed in matrix form, it is easy to construct the matrix $W$ corresponding to the embedding: Each path in the embedding shows up as a column in $W$. The rows in $W$ correspond to the edges in $V$; if an edge in $V$ is in the path, a $+1$ or $-1$ occurs in that row of $W$, depending on whether the direction of the edge in $V$ is in the same or opposite direction of the path, respectively. (Again, entries in $W$ can be scaled to deal with weighted graphs.) In this representation, $U = VW$ expresses the mapping defined by the embedding.

Properties of the paths were used to compute bounds. This was typically done in combinatorial rather than algebraic terms. One method, suggested in Vaidya's work (see the section, "An Analogy with Resistive Networks," on p. 6 of [13]) and developed by Gremban and Miller (see the discussion starting with the last two paragraphs on p. 65 and continuing to the start of section 4.5 on the next page of [5]), involved summing the congestions (defined for each edge in the graph embedded into the number of paths in the embedding incident to that edge) along each path. The maximum sum taken over all paths provided an upper bound on $\lambda_{\max}(B^{-1}A)$.

This method was also applied in a line of research that considered bounding the smallest nontrivial eigenvalue of a Laplacian or related matrix (the earliest work in this direction was aimed at bounding the second largest eigenvalues of time-reversible Markov chains in order to bound the mixing time for random walks [10, 12]). This typically involved embedding the complete graph into a Laplacian or a generalization of the Laplacian that allowed weighted edges. Later work involved a further generalization that included Dirichlet boundary conditions (these conditions resulted in entries on the diagonal that exceeded the sum of the weights of the incident edges). For such matrices, a star was embedded instead of a clique.

Analysis using path congestions and related techniques initially did not express the embedding in matrix form. However, some work along these lines developed an algebraic representation of embeddings. Kahale [11] looked at computing lower bounds on the smallest nontrivial eigenvalue of a Laplacian using a method that assigned a length to each path, then looked for the edge that had the greatest sum of the lengths of the incident paths. He computed embedding properties in terms of $|W|$, the absolute value of the matrix $W$ representing the embedding, and expressed the best bound that could be computed for any embedding in terms of $\lambda_{\max}(|W|^T|W|)$. Guattery, Leighton, and Miller [7] formulated the path resistance method, an extension of the sum-of-congestions method applied to lower bounds of smallest nontrivial Laplacian eigenvalues. They showed that, given an embedding $W$, the best possible bound computed using the path resistance method was the same as the best possible bound computed using Kahale's edge-length method and that the value $\lambda_{\max}(|W||W|^T) = \lambda_{\max}(|W|^T|W|)$ was a term in the expression for this best bound.

Guattery and Miller [8] made the observation that including directions in embeddings typically improved the best possible lower bounds on smallest nontrivial eigenvalues of generalized Laplacians that could be derived from the embeddings. Guattery and Miller allowed multiple paths in embedding each edge and they also kept the signs corresponding to direction in the embedding matrix $W$, which corresponds to working with $W$ rather than $|W|$. At this point the notion of embedding has been generalized to the point that any matrix $W$ such that $U = VW$ can be viewed as a [illegible] of $U$ into $V$, where $U$ and $V$ are vertex-edge incidence matrices. Guattery and Miller also proved that there existed a particular embedding $W_{cf}$ (referred to as the [illegible] such that the smallest nontrivial eigenvalue could be computed exactly in terms of $\lambda_{\max}(W_{cf}W_{cf}^T)$ for that embedding. They also showed that $W_{cf}$ was a factor of the inverse for this embedding applied to the case of a generalized Laplacian with a Dirichlet boundary condition.

Guattery [6] extended these ideas to all Hermitian matrices. In particular, he considered what he called the [illegible] $W_{cf}$ of the orthogonal projector onto the range of the columns of $B = VDV^*$ (where $D$ is a diagonal matrix with nonzeros on the diagonal) into $V$. He showed that the pseudoinverse of $B$ could be expressed as $W_{cf}D^{-1}W_{cf}^*$. The formulation is still in terms of a slightly generalized vertex-edge incidence matrix although, as shown in section 6, the ideas can be generalized to any factorization $VDV^*$.

The results in this paper tie together ideas from the algebraic and combinatorial (path embedding) views of support theory, and express them in a common notation.

**4. The main theorem.** Consider the matrix pencil $(A, B)$, where $A$ and $B$ are $n \times n$ complex, Hermitian matrices. Assume that there exist matrices $U$ and $V$ such that $A = UD_AU^*$ and $B = VD_BV^*$, where $U$ is $n \times m$, $V$ is $n \times k$, and $D_B$ and $D_A$ are invertible Hermitian matrices with dimensions $k \times k$ and $m \times m$, respectively. The main theorem can be stated as follows.

THEOREM 4.1 (main theorem). [illegible] $(A, B)$ [illegible] $U$ [illegible] $B$ [illegible] $W_{opt}$ [illegible] $VW_{opt} = U$ [illegible] $W_{opt}^* D_B^{-1} W_{opt} D_A$ [illegible] $B^+A$ [illegible]

Before proving the main theorem, it is helpful to lay out some supporting lemmas. Define the $(n + k) \times (n + k)$ block matrix $M$ as follows:

$$M = \begin{bmatrix} D_B^{-1} & V^* \\ V & 0 \end{bmatrix}. \tag{4.1}$$

The proof of the main theorem is based on solutions to the following block system:

$$\begin{bmatrix} D_B^{-1} & V^* \\ V & 0 \end{bmatrix} \begin{bmatrix} W_{opt} \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ U \end{bmatrix}. \tag{4.2}$$

The $k \times m$ matrix $W_{opt}$ is of particular interest. The following lemma gives necessary and sufficient conditions for the existence of a solution, and hence immediately implies the existence of $W_{opt}$:

LEMMA 4.2. [illegible] if and only if [illegible] $U$ [illegible] $B$

[illegible]. We can apply Gaussian elimination to $M$ as follows:

$$\begin{bmatrix} I & 0 \\ -VD_B & I \end{bmatrix} \begin{bmatrix} D_B^{-1} & V^* \\ V & 0 \end{bmatrix} \begin{bmatrix} W_{opt} \\ Z \end{bmatrix} = \begin{bmatrix} I & 0 \\ -VD_B & I \end{bmatrix} \begin{bmatrix} 0 \\ U \end{bmatrix}$$

to get the following reduced system:

$$\begin{bmatrix} D_B^{-1} & V^* \\ 0 & -B \end{bmatrix} \begin{bmatrix} W_{opt} \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ U \end{bmatrix}. \tag{4.3}$$

Proving the result for the reduced matrix is sufficient because it has the same solutions as the original system.

Assuming that a solution to the reduced system exists, there must be a $Z$ such that $-BZ = U$. That is, a solution exists only if the columns of $U$ are in the range of $B$.

Assuming that the columns of $U$ are in the range of $B$, then a $Z$ that satisfies $-BZ = U$ exists. For the block system to have a solution, we must also have a solution to the equation $D_B^{-1}W_{opt} + V^*Z = 0$. The solution exists if there is a $W_{opt}$ such that $W_{opt} = -D_B V^* Z$. Since the matrices $D_B$, $V^*$ and $Z$ exist by assumption, a solution does exist and the lemma holds. ☐

It is possible that there is more than one solution. In some arguments below we require a solution that is orthogonal to the null space of $M$; where this is the case, it is noted explicitly.

Theorem 1.3.20 from [9] is useful in our arguments. We restate (in slightly revised form) the pertinent result below.

THEOREM 4.3. [illegible] $Y$ [illegible] $r \times s$ [illegible] $Z$ [illegible] $s \times r$ [illegible] $r \leq s$ [illegible] $ZY$ [illegible] $YZ$ [illegible] $s - r$ [illegible] $0$

We can now prove the main theorem.

[illegible] Given the factorizations assumed in the theorem statement, we can construct the block system in (4.2). As in Lemma 4.2, we can apply Gaussian elimination to get the reduced system from (4.3). Assume that the columns of $U$ are in the range of $B$. By Lemma 4.2, a solution to the reduced system exists. That immediately yields the equation

$$-BZ = U.$$

Multiplying through on both sides by $-B^+$ gives

$$P_{R(B^+)}Z = -B^+U.$$

It is a fact that $P_{R(B^+)} = P_{R(B^*)}$ (see p. 10 of [4]). Thus we can rewrite the equation as follows:

$$P_{R(B^*)}Z = -B^+U. \tag{4.4}$$

The assumption that the columns of $U$ are in the range of $B$ also implies that $U = P_{R(B)}U$. Taking the conjugate transpose gives $U^* = U^*P_{R(B)}^* = U^*P_{R(B)}$, the last equality following because orthogonal projectors are Hermitian by definition. Because $B$ is Hermitian, $P_{R(B)} = P_{R(B^*)}$. Thus we have the following equation:

$$U^* = U^*P_{R(B^*)}. \tag{4.5}$$

From the original block system we have that

$$D_B^{-1}W_{opt} = -V^*Z \tag{4.6}$$

and also that, after taking the conjugate transpose,

$$W_{opt}^*V^* = U^*. \tag{4.7}$$

Applying (4.6), (4.7), (4.5), and (4.4) in turn, we have

$$\begin{aligned} W_{opt}^*D_B^{-1}W_{opt}D_A &= -W_{opt}^*V^*ZD_A \\ &= -U^*ZD_A \\ &= -U^*P_{R(B^*)}ZD_A \\ &= U^*B^+UD_A. \end{aligned} \tag{4.8}$$

We can apply Theorem 4.3 to show that each nonzero eigenvalue of $U^*B^+UD_A$ is an eigenvalue of $B^+A$ and vice-versa.

If $m > n$, we apply Theorem 4.3 to $B^+A = B^+UD_AU^*$. Noting that $B^+UD_A$ is $n \times m$ and $U^*$ is $m \times n$, we immediately have that $B^+A$ has the same nonzero eigenvalues (counting multiplicity) as $U^*B^+UD_A$, plus $m - n$ additional zero eigenvalues.

If $m \leq n$, we apply Theorem 4.3 to $U^*B^+UD_A$. We immediately have that $B^+UD_AU^* = B^+A$ has the same nonzero eigenvalues as $U^*B^+UD_A$ (counting multiplicity), plus $n - m$ additional zero eigenvalues if $n$ is strictly greater than $m$.

This completes the proof. ☐

As a consequence of Theorem 4.1, we have the additional result.

THEOREM 4.4. [illegible] $U \in \mathbb{R}^{n\times k}$ [illegible] $V \in \mathbb{R}^{n\times p}$ [illegible] $W$ [illegible] $VW = U$ [illegible] $W^TW$ [illegible] $(VV^T)^+UU^T$

[illegible] The result follows immediately from Theorem 4.1, with $D_B = I$ and $D_A = I$. ☐

Note that in this case, $D_B = I$ in the system from (4.2), so in the solution orthogonal to the null space of $M$ from (4.1), $W$ is the solution to the equation $VW = U$ with the minimum two-norm. This is consistent with Theorem 3.1 (see Theorem 4.5 from [3]).

## 5. The non-Hermitian case.

A related theorem can be proved for invertible, complex non-Hermitian matrices of the form $A = EF^*$, where $A$ is $n \times n$, and $E$ and $F$ are $n \times m$, with $m \geq n$. Consider the following two block systems: one for $A$:

$$\begin{bmatrix} I_m & F^* \\ E & 0 \end{bmatrix} \begin{bmatrix} W \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ I_n \end{bmatrix}, \tag{5.1}$$

and one for $A^* = FE^*$:

$$\begin{bmatrix} I_m & E^* \\ F & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 0 \\ I_n \end{bmatrix}. \tag{5.2}$$

With respect to these systems and the assumption that $A$ is invertible, we have the following theorem.

THEOREM 5.1. $X^*W = A^{-1}$

[illegible] The system in (5.1) yields the equations

$$W = -F^*Z$$

and

$$EW = I_n, \tag{5.3}$$

which, when combined, yield the equation

$$-EF^*Z = I_n,$$

which implies that

$$-Z = A^{-1}.$$

Likewise the system in (5.2) yields the equations

$$X = -E^*Y, \tag{5.4}$$

$$FX = I_n,$$

$$-FE^*Y = I_n,$$

and

$$-Y = (A^*)^{-1}. \tag{5.5}$$

Hence

$$\begin{aligned} X^*W &= -Y^*EW && \text{(by (5.4))} \\ &= -Y^* && \text{(by (5.3))} \\ &= A^{-1} && \text{(by (5.5))}. \quad \square \end{aligned}$$

**6. Generalized embeddings and pseudoinverses.** The main theorem can also be applied to producing pseudoinverses. Guattery, in a technical report [6], showed that by applying a generalized version of embedding to the orthogonal projector onto the range space of any symmetric matrix, it is possible to generate factors of that matrix's pseudoinverse. He also showed how to extend this result to all Hermitian matrices by splitting them into a real and an imaginary part and working with a system twice as big. Theorem 4.1 allows us to prove the results from this technical report in a simpler way. The new proof covers the Hermitian case directly.

Let $B$ be a Hermitian matrix, with $B = VDV^*$, $D$ Hermitian, and invertible. Let $\begin{bmatrix} W \\ Z \end{bmatrix}$ be the solution to the system

$$\begin{bmatrix} D^{-1} & V^* \\ V & 0 \end{bmatrix} \begin{bmatrix} W \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ P_{R(B)} \end{bmatrix}, \tag{6.1}$$

where $P_{R(B)}$ is the orthogonal projector onto the range space of $B$. The matrix $W$ here corresponds to what Guattery termed a [illegible].

The following two theorems are direct consequences of Theorem 4.1 and its proof.

THEOREM 6.1. $W^*D^{-1}W = B^+$

[illegible]. Viewing the system above in terms of the system in Theorem 4.1, we can make the substitutions $U = P_{R(B)}$, $D_B = D$, and $D_A = I_n$. Substituting these values in (4.8) from the proof of Theorem 4.1 gives

$$W^*D^{-1}W = P^*_{R(B)}B^+P_{R(B)}.$$

Note that $P_{R(B^+)} = P_{R(B^*)}$ (see [4]). Recall that $P_{R(B)} = P_{R(B^*)}$ since $B$ is Hermitian and that $P^*_{R(B)} = P_{R(B)}$ because orthogonal projectors are Hermitian by definition. These equations also imply that $P_{R(B^+)} = P_{R(B)}$. Hence we have the following:

$$P^*_{R(B)}B^+P_{R(B)} = P_{R(B^+)}B^+P_{R(B)} = B^+P_{R(B)} = B^+.$$

The last equality follows from the fact that $B^+$ is Hermitian, so $R(B^+)$ is a reducing subspace. Proposition 0.2.3 from [4] thus gives us that

$$P_{R(B^+)}B^+ = B^+P_{R(B^+)} = B^+.$$

Since $P_{R(B^+)} = P_{R(B)}$ (as noted above), this proves the theorem. $\square$

In the proof above, note that the factor embedded, $P_{R(B)}$, is also equal to the symmetric product of the factor and its conjugate transpose. This is done to ensure $W$ has the proper dimensions to serve as a factor of the pseudoinverse.

One possibility hidden in the argument is that the rank of $B$ may be less than the rank of $V$. In such cases the matrix $D$ projects part of the range of $V^*$ into the null space of $V$. When this does not happen, however, we can prove an interesting property. Let matrices $V$, $D$, and $B$ be defined as for Theorem 6.1. Let $\left[\begin{smallmatrix} W \\ Z \end{smallmatrix}\right]$ be a solution to (6.1) that is orthogonal to the null space of the block matrix. We have the following theorem.

THEOREM 6.2. rank(B) = rank(V) $-Z = B^{+}$

. Assume that the order of $D$ is $k$ and that $V$ is an $n \times k$ matrix. We can apply Gaussian elimination to the system in (6.1) as follows:

$$\begin{bmatrix} I & 0 \\ -VD & I \end{bmatrix} \begin{bmatrix} D^{-1} & V^{*} \\ V & 0 \end{bmatrix} \begin{bmatrix} W \\ Z \end{bmatrix} = \begin{bmatrix} I & 0 \\ -VD & I \end{bmatrix} \begin{bmatrix} 0 \\ P_{R(B)} \end{bmatrix}$$

to get the following block upper triangular system:

$$\begin{bmatrix} D^{-1} & V^{*} \\ 0 & -B \end{bmatrix} \begin{bmatrix} W \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ P_{R(B)} \end{bmatrix},$$

The rank of the block matrix (referred to as $M$ below) is greater than or equal to the sum of the ranks of $B$ and $D^{-1}$ (see, e.g., [9]). Since $D^{-1}$ has full rank, this also implies that the dimension of the null space of $M$ is less than or equal to the size of the null space of $B$.

Note that we have $-BZ = P_{R(B)}$.

If $B$ is nonsingular, then the result follows immediately: $P_{R(B)} = I$ and $-BZ = I$. By the uniqueness of the inverse, $Z = -B^{-1}$.

If $B$ has less than full rank, multiplying through on both sides by $B^{+}$ gives the equation

$$-P_{R(B)}Z = B^{+}P_{R(B)} = B^{+},$$

where the last equality follows by the argument given in Theorem 6.1. The theorem therefore holds if the columns of $Z$ are in the range of $B$.

Since $B = VDV^{*}$, the null space of $V^{*}$ is contained in the null space of $B$. The condition in the theorem statement that the ranks of $B$ and $V$ are the same thus implies that the null spaces of $B$ and $V^{*}$ are the same.

We can construct a basis for the null space of the block matrix $M$ as follows: Assume the size of the null space of $B$ is $j \geq 1$. Construct an orthogonal basis for the null space of $B$ consisting of vectors $\vec{v_1} \cdots \vec{v_j}$. Consider the $j$ vectors of the form $\vec{w_i} = \left[\begin{smallmatrix} \vec{0}_k \\ \vec{v_i} \end{smallmatrix}\right]$, where $\vec{0}_k$ is the column vector consisting of $k$ zeros. These vectors are clearly orthogonal. Because $B$ and $V^{*}$ have the same null spaces, each such vector is in the null space of $M$. And because (as noted above) the size of the null space of $M$ is less than or equal to size of the null space of $B$, the $\vec{w_i}$'s span the null space of $M$, and hence form a basis for it.

By assumption, the solution $\left[\begin{smallmatrix} W \\ Z \end{smallmatrix}\right]$ is orthogonal to the null space of $M$. By the structure of the vectors in the null space of $M$, this means that $Z$ is orthogonal to the null space of $B$. Hence $Z$ is in the range of $B$, and the theorem holds. ☐

**7. Approximate inverse preconditioning.** In preconditioning linear systems $Ax = b$, one often constructs a preconditioner $B \approx A$. In the iterative method, one has to solve for (invert) $B$. An alternative is to algebraically construct a matrix $M$ such that $MA \approx I$. Since $M$ approximates the inverse of $A$, one only needs to multiply by $M$ in the iterative method, which has certain advantages (e.g., in parallel computing).

Many strategies have been proposed for constructing approximate inverses; see, e.g., [1]. One practical condition is that $M$ must be sparse. Some try to minimize $\|MA - I\|$ (with certain sparsity constraints) directly, while others construct $M$ as the product of two triangular factors.

An interesting open question is whether our Theorem 6.1 can be used as a starting point for constructing approximate inverses. This theorem provides a novel factorization for the inverse of a symmetric matrix. Are there cases in which we could compute an inexpensive approximation $\tilde{W}$ to $W$? Then $\tilde{W}^* D^{-1} \tilde{W}$ would be an approximation to the inverse.

The key issues to be settled are whether there exist classes of matrices for which a good, sparse approximation to $W$ exists. Additionally, there is the question of which factorization of $A$ (if any) can be used to find good approximate factors, and how the approximation is constructed based on such a factorization.

It is also interesting to consider a similar strategy applied to nonsymmetric matrices using Theorem 5.1. Can good approximations to the factors $X$ and $W$ of our nonsymmetric matrix be generated by approximately solving (5.1) or (5.2)?

We remark that approximate inverses of this type are in factored form but the factors may be rectangular (nonsquare), which would make such an approach distinctively different from existing factored approximate inverses.

## REFERENCES

[1] M. Benzi and M. Tuma, *A comparative study of sparse approximate inverse preconditioners*, Appl. Numer. Math., 30 (1999), pp. 305–340.

[2] M. Bern, J. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, *Support-graph preconditioners*, SIAM J. Matrix Anal. Appl., submitted.

[3] E. G. Boman and B. Hendrickson, *Support theory for preconditioning*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 694–717.

[4] S. L. Campbell and C. D. Meyer, Jr., *Generalized Inverses of Linear Transformations*, Corrected reprint of the 1979 original, Dover, New York, 1991.

[5] K. D. Gremban, *Combinatorial Preconditioners for Sparse, Symmetric, Diagonally Dominant Linear Systems*, Ph.D. thesis and CMU CS Technical report CMU-CS-96-123, Carnegie Mellon University, Pittsburgh, PA.

[6] S. Guattery, *Graph Embeddings, Symmetric Real Matrices, and Generalized Inverses*, Technical report 98-34, ICASE, NASA Langley, Hampton, VA, 1998.

[7] S. Guattery, F. T. Leighton, and G. L. Miller, *The path resistance method for bounding the smallest nontrivial eigenvalue of a Laplacian*, Combin. Probab. Comput., 8 (1999), pp. 441–460.

[8] S. Guattery and G. L. Miller, *Graph embeddings and Laplacian eigenvalues*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 703–723.

[9] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[10] M. Jerrum and A. Sinclair, *Approximating the permanent*, SIAM J. Comput., 18 (1989), pp. 1149–1178.

[11] N. Kahale, *A semidefinite bound for mixing rates of Markov chains*, in Integer Programming and Combinatorial Optimization, Lecture Notes in Comput. Sci. 1084, 1996, pp. 190–203.

[12] A. Sinclair and M. Jerrum, *Approximate counting, uniform generation and rapidly mixing Markov chains*, Inform. and Comput., 82 (1989), pp. 93–133.

[13] P. M. Vaidya, *Solving Linear Equations with Symmetric Diagonally Dominant Matrices by Constructing Good Preconditioners*, IMA Workshop on Graph Theory and Sparse Matrix Computation, 1991.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 606–625

# REGULARIZATION IN REGRESSION WITH BOUNDED NOISE: A CHEBYSHEV CENTER APPROACH*

AMIR BECK[†] AND YONINA C. ELDAR[‡]

**Abstract.** We consider the problem of estimating a vector $\mathbf{z}$ in the regression model $\mathbf{b} = \mathbf{A}\mathbf{z}+\mathbf{w}$, where $\mathbf{w}$ is an unknown but bounded noise. As in many regularization schemes, we assume that an upper bound on the norm of $\mathbf{z}$ is available. To estimate $\mathbf{z}$ we propose a relaxation of the Chebyshev center, which is the vector that minimizes the worst-case estimation error over all feasible vectors $\mathbf{z}$. Relying on recent results regarding strong duality of nonconvex quadratic optimization problems with two quadratic constraints, we prove that in the *complex domain* our approach leads to the exact Chebyshev center. In the real domain, this strategy results in a "pretty good" approximation of the true Chebyshev center. As we show, our estimate can be viewed as a Tikhonov regularization with a special choice of parameter that can be found efficiently by solving a convex optimization problem with two variables or a semidefinite program with three variables, regardless of the problem size. When the norm constraint on $\mathbf{z}$ is a Euclidean one, the problem reduces to a single-variable convex minimization problem. We then demonstrate via numerical examples that our estimator can outperform other conventional methods, such as least-squares and regularized least-squares, with respect to the estimation error. Finally, we extend our methodology to other feasible parameter sets, showing that the total least-squares (TLS) and regularized TLS can be obtained as special cases of our general approach.

**Key words.** Chebyshev center, nonconvex quadratic optimization, strong duality, bounded error estimation

**AMS subject classifications.** 90C20, 90C22, 90C26, 65F30

**DOI.** 10.1137/060656784

## 1. Introduction.

Many problems in data fitting and estimation give rise to a system of linear equations $\mathbf{A}\mathbf{z} \approx \mathbf{b}$ where the right-hand side $\mathbf{b}$ is contaminated by noise. More specifically, we consider the linear model

$$\mathbf{b} = \mathbf{A}\mathbf{z} + \mathbf{w}, \tag{1}$$

where $\mathbf{A} \in \mathbb{F}^{m\times n}$ is the model matrix, $\mathbf{b} \in \mathbb{F}^m$ is the observation vector, $\mathbf{w} \in \mathbb{F}^m$ is the unknown noise (or "error"), and $\mathbf{z} \in \mathbb{F}^n$ is the unknown parameter vector. Here $\mathbb{F}$ denotes either the real number field $\mathbb{R}$ or the complex number field $\mathbb{C}$. Given the observation $\mathbf{b}$, we seek an estimator $\hat{\mathbf{z}}$ of $\mathbf{z}$ that is close in some sense to $\mathbf{z}$. This estimation problem arises in a large variety of areas in science and engineering, e.g., communication, economics, signal processing, seismology, and control.

The celebrated least-squares (LS) approach [5, 18] to estimating $\mathbf{z}$ in the model (1) is to seek the vector $\hat{\mathbf{z}}_{\mathrm{LS}}$ that minimizes the norm of the data error $\|\mathbf{A}\hat{\mathbf{z}} - \mathbf{b}\|^2$, where $\|\mathbf{v}\|$ stands for the Euclidean norm of the vector $\mathbf{v}$. When $\mathbf{A}$ has full column rank, $\hat{\mathbf{z}}_{\mathrm{LS}}$ is given by

$$\hat{\mathbf{z}}_{\mathrm{LS}} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b}. \tag{2}$$

---

[*] Received by the editors April 8, 2006; accepted for publication (in revised form) by J. G. Nagy December 7, 2006; published electronically May 1, 2007.
http://www.siam.org/journals/simax/29-2/65678.html

[†] Department of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel (becka@ie.technion.ac.il).

[‡] Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (yonina@ee.technion.ac.il).

In practical situations the matrix $\mathbf{A}$ is often ill-conditioned—for example, when the system is obtained via discretization of ill-posed problems such as integral equations of the first kind (see, e.g., [16] and references therein). In these cases the LS solution might give poor results with respect to the estimation error. A well-established approach for stabilizing the LS estimate is to incorporate prior information on the true parameter vector $\mathbf{z}$ into the optimization problem (2) by adding a quadratic constraint:

$$\hat{\mathbf{z}}_{\mathrm{RLS}} \in \operatorname*{argmin}_{\mathbf{z}\in\mathbb{F}^n}\{\|\mathbf{A}\mathbf{z}-\mathbf{b}\|^2 : \|\mathbf{L}\mathbf{z}\|^2 \leq \eta\}. \tag{3}$$

The matrix $\mathbf{L}$ is often chosen as the identity, or as a discrete approximation of some derivative operator (see [5, 16]). The resulting estimator is referred to as the [illegible] LS (RLS) estimator [5]. It is well known that $\hat{\mathbf{z}}_{\mathrm{RLS}}$ is either equal to the LS solution when $\|\mathbf{L}\mathbf{z}_{\mathrm{LS}}\|^2 \leq \eta$ or given by $\hat{\mathbf{z}}_{\mathrm{RLS}} = \mathbf{z}_\lambda$, where $\mathbf{z}_\lambda$ satisfies the generalized normal equations [5]

$$(\mathbf{A}^*\mathbf{A} + \lambda\mathbf{L}^*\mathbf{L})\mathbf{z}_\lambda = \mathbf{A}^*\mathbf{b}. \tag{4}$$

The parameter $\lambda$ is determined by the secular equation $\|\mathbf{L}\mathbf{z}_\lambda\|^2 = \eta$. Therefore, the RLS solution is a [illegible] estimator [28] with a choice of regularization parameter $\lambda$ that takes into account the norm constraint $\|\mathbf{L}\mathbf{z}\|^2 \leq \eta$.

It is important to note that both the LS and the RLS strategies are based on minimizing the data error. However, in an estimation context, typically we would like to minimize the squared [illegible] $\|\hat{\mathbf{z}} - \mathbf{z}\|^2$. When the noise $\mathbf{w}$ in (1) is assumed to be random with zero mean and known covariance matrix, the squared estimation error will also be a random variable. Using the known statistics of $\mathbf{w}$, the average squared estimation error, referred to as the mean-squared error (MSE), can be computed. Several different strategies based on the MSE have been recently proposed [25, 9, 8, 2, 7]. These methods consider [illegible] estimates of $\mathbf{z}$ and assume knowledge of the statistics of $\mathbf{w}$.

**1.1. Bounded error estimation.** In some scenarios, the distribution of the noise might not be known exactly (or at all). There are also cases where the noise is not inherently random (for example, in problems resulting from quantizing a continuous-time signal). This leads to the [illegible] approach which deals with unknown but bounded noise (see, e.g., [19] and the survey papers [21, 24]). In this paper we adopt the bounded error methodology and assume that the noise is norm-bounded $\|\mathbf{w}\|^2 \leq \rho$. As in the RLS strategy, in order to obtain a stable solution, we further restrict $\mathbf{z}$ to have weighted bounded norm.

The first stage in the deterministic bounded error approach is to construct all admissible solutions to the linear system (1); for this reason this approach is also referred to as [illegible] [21]. In our setting, the feasible parameter set (FPS) is given by the intersection of two ellipsoids[1]:

$$\mathrm{FPS} = \{\mathbf{z} \in \mathbb{F}^n : \|\mathbf{L}\mathbf{z}\|^2 \leq \eta, \|\mathbf{A}\mathbf{z}-\mathbf{b}\|^2 \leq \rho\}. \tag{5}$$

The second step is to choose a central representative of the FPS. A popular choice is the Chebyshev center [29], which is defined as the solution $\hat{\mathbf{z}}$ to the following min-max problem:

$$\min_{\hat{\mathbf{z}}\in\mathbb{F}^n} \max_{\mathbf{z}\in\mathrm{FPS}} \|\mathbf{z} - \hat{\mathbf{z}}\|^2. \tag{6}$$

---

[1] Note that here the norm bound $\|\mathbf{w}\|^2 \leq \rho$ translates to $\|\mathbf{A}\mathbf{z}-\mathbf{b}\|^2 \leq \rho$.

Geometrically, the Chebyshev center is the center of the minimum radius ball enclosing the FPS; the optimal value of (6) is the squared radius of the minimal ball enclosing the set. This is illustrated in Figure 1 with the filled area being the intersection of two ellipsoids. The dotted circle is the minimum inscribing circle of the intersection of the ellipsoids.

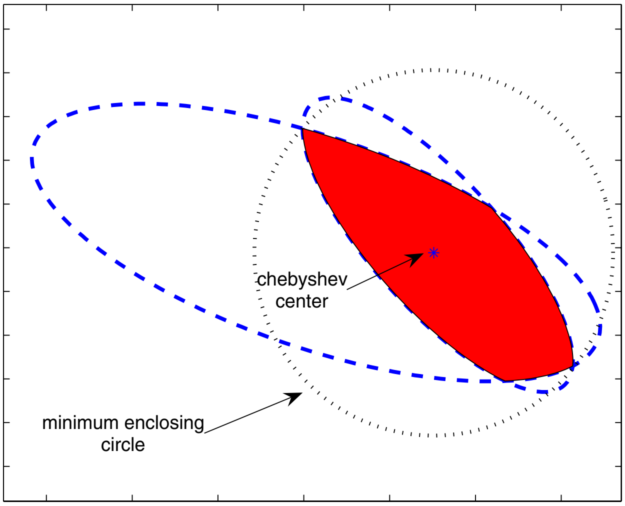


FIG. 1. *The Chebyshev center of the intersection of two ellipsoids.*

The Chebyshev center of the FPS gives the best worst-case estimation error over the set. Thus, it is aimed at optimizing an objective that depends on the estimation error rather than the data error. In section 5 we demonstrate by simulations that an estimator based on the Chebyshev center typically performs worse than the LS and RLS approaches with respect to the data error; however, it appears to perform [illegible] better in terms of the estimation error even when only loose bounds on the norm of the noise ($\rho$) are known. Thus, this approach can improve the estimation error without requiring much more knowledge than the RLS strategy.

Finding a Chebyshev center of a convex set is, in general, a hard problem. Two exceptions are the case where the set is polyhedral and the enclosing ball is the $l_\infty$ ball [20], and the case when the set is finite (see, e.g., [30] and references therein).

The Chebyshev center problem (6) we tackle in this paper is seemingly hard. To better understand the intrinsic difficulty of this min-max problem, note that the inner maximization problem is a [illegible] quadratic optimization problem. However, relying on some recent strong duality results derived in the context of quadratic optimization [1], we will show that despite the nonconvexity of the problem, it can be solved efficiently when $\mathbb{F} = \mathbb{C}$. The same approach can be used when $\mathbb{F} = \mathbb{R}$ to develop an approximation of the Chebyshev center. Simulation results show that this approximation is pretty good in the sense that it yields favorable estimation performance.

**1.2. Paper layout and main results.** A review of the relevant optimization concepts and the strong duality results of [1] is given in section 2. These results are then used in section 3 to reduce the problem of finding the Chebyshev center of the intersection of two level sets of quadratic functions[2] with $\mathbb{F} = \mathbb{C}$ to a convex optimization problem in only two variables. This problem can also be recast as a semidefinite program (SDP) involving linear matrix inequality (LMI) constraints, with three variables.

In section 4 we present the relaxed Chebyshev center (RCC) estimator, which is exactly the Chebyshev center of the FPS in the case $\mathbb{F} = \mathbb{C}$ under strict feasibility constraints. We show that the RCC, like the RLS solution, is a Tikhonov estimator. However, in the RCC approach, as opposed to the RLS method, the regularization parameter is chosen to account for [illegible] constraints defining the FPS. Furthermore, it is designed to minimize an estimation error rather than a data error. We also show that when considering the FPS with a Euclidean norm constraint on $\mathbf{z}$ (i.e., $\mathbf{L} = \mathbf{I}$), the problem reduces to a convex optimization problem with a single variable.

Section 5 presents numerical examples demonstrating the effectiveness of the RCC strategy. We also compare two methods for evaluating the RCC estimator: an implementation of the ellipsoid method [3] (described in full detail in Appendix A) and a standard interior point method applied to the resulting SDP. We show both theoretically and numerically that in our problem the ellipsoid method is more efficient.

Finally, in section 6, we extend our approach to several related problems, and show that the total LS (TLS) [13, 17] and regularized TLS (RTLS) estimators [12] can be viewed as special cases of our general methodology.

**1.3. Notation.** Throughout the paper, the following notation is used: vectors are denoted by boldface lowercase letters, e.g., $\mathbf{y}$, and matrices by boldface uppercase letters, e.g., $\mathbf{A}$. The $i$th component of a vector $\mathbf{y}$ is written as $y_i$, and $\hat{(\cdot)}$ is an estimated vector. The identity matrix is denoted by $\mathbf{I}$. The real and imaginary parts of scalars, vectors, or matrices are written as $\Re(\cdot)$ and $\Im(\cdot)$. For a matrix $\mathbf{A}$, $\mathbf{A}^*, \mathbf{A}^T, \mathbf{A}^\dagger$, and $\mathcal{R}(\mathbf{A})$ are the Hermitian conjugate, transpose, Moore-Penrose generalized inverse [14], and image space. For a square symmetric matrix, $\lambda_{\min}(\mathbf{A})$ is the minimum eigenvalue of $\mathbf{A}$. Given two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \succ \mathbf{B}$ ($\mathbf{A} \succeq \mathbf{B}$) means that $\mathbf{A} - \mathbf{B}$ is positive definite (semidefinite). The value of the optimal objective function of an optimization problem

$$(P): \min / \max\{f(\mathbf{x}) : \mathbf{x} \in C\}$$

is denoted by $\mathrm{val}(P)$. For simplicity, instead of inf/sup we use min/max; however this does not mean that we assume that the optimum is attained and/or finite.

## 2. Quadratically constrained quadratic programs: A review.

Our goal is to find the Chebyshev center of the FPS (5). The difficulty is that the inner maximization in (6)

$$\max_{\mathbf{z} \in \mathrm{FPS}} \|\mathbf{z} - \hat{\mathbf{z}}\|^2$$

is not convex. In this section, we summarize prior results concerning the minimization of a general quadratic form subject to quadratic constraints. We will then show, in sections 3 and 4, how these results can be applied in order to solve (6).

---

[2] This is a more general form than a set that is an intersection of two ellipsoids.

Consider the general form quadratically constrained quadratic problem

$$(\mathrm{QP}_m) \quad \min_{\mathbf{z}\in\mathbb{F}^n}\{f_0(\mathbf{z}) : f_i(\mathbf{z}) \le 0, i = 1,\dots,m\},$$

where $m$ denotes the number of constraints, and

$$f_i(\mathbf{z}) = \mathbf{z}^*\mathbf{A}_i\mathbf{z} + 2\Re(\mathbf{b}_i^*\mathbf{z}) + c_i$$

with $\mathbf{A}_i = \mathbf{A}_i^* \in \mathbb{F}^{n\times n}, \mathbf{b}_i \in \mathbb{F}^n$, and $c_i \in \mathbb{R}$ for $i = 0,\dots,m$. Note that in the case $\mathbb{F} = \mathbb{R}$, the quadratic functions $f_i(\mathbf{z})$ can be written as $\mathbf{z}^T\mathbf{A}_i\mathbf{z} + 2\mathbf{b}_i^T\mathbf{z} + c_i$.

The problem $(\mathrm{QP}_m)$ is in general not convex since $\mathbf{A}_i$ are not necessarily positive semidefinite. The Lagrangian dual of $(\mathrm{QP}_m)$ is the maximization problem [4, 26]

$$\max_{\boldsymbol{\alpha}}\{q(\boldsymbol{\alpha}) : \boldsymbol{\alpha} \ge 0\}, \tag{7}$$

where $q(\boldsymbol{\alpha})$ is the dual objective function defined by

$$q(\boldsymbol{\alpha}) = \min_{\mathbf{z}\in\mathbb{F}^n}\left\{f_0(\mathbf{z}) + \sum_{i=1}^m \alpha_i f_i(\mathbf{z})\right\}.$$

The function $q(\boldsymbol{\alpha})$ can also be written in the form

$$q(\boldsymbol{\alpha}) = \max_{\lambda}\left\{\lambda : f_0(\mathbf{z}) + \sum_{i=1}^m \alpha_i f_i(\mathbf{z}) \ge \lambda \text{ for every } \mathbf{z} \in \mathbb{F}^n\right\}. \tag{8}$$

To obtain a more convenient representation of $q(\boldsymbol{\alpha})$ we exploit the following well-known lemma.

LEMMA 2.1 (see [3, p. 163]). [illegible] $g : \mathbb{F}^n \to \mathbb{R}$ [illegible] $g(\mathbf{z}) = \mathbf{z}^*\mathbf{A}\mathbf{z} + 2\Re(\mathbf{b}^*\mathbf{z}) + c$ [illegible] $\mathbf{A} = \mathbf{A}^* \in \mathbb{F}^{n\times n}, \mathbf{b} \in \mathbb{F}^n$ [illegible] $c \in \mathbb{R}$ [illegible]

(i) $g(\mathbf{z}) \ge 0$ [illegible] $\mathbf{z} \in \mathbb{F}^n$

(ii) $\begin{pmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^* & c \end{pmatrix} \succeq \mathbf{0}$

Applying Lemma 2.1 to (8), we can represent $q(\boldsymbol{\alpha})$ as

$$q(\boldsymbol{\alpha}) = \max_{\lambda}\left\{\lambda : \begin{pmatrix} \mathbf{A}_0 & \mathbf{b}_0 \\ \mathbf{b}_0^* & c_0 - \lambda \end{pmatrix} + \sum_{i=1}^m \alpha_i \begin{pmatrix} \mathbf{A}_i & \mathbf{b}_i \\ \mathbf{b}_i^* & c_i \end{pmatrix} \succeq \mathbf{0}\right\}.$$

The dual problem (7) then becomes

$$(\mathrm{D}_m) \quad \max_{\alpha_i\ge 0,\lambda}\left\{\lambda : \begin{pmatrix} \mathbf{A}_0 & \mathbf{b}_0 \\ \mathbf{b}_0^* & c_0 - \lambda \end{pmatrix} + \sum_{i=1}^m \alpha_i \begin{pmatrix} \mathbf{A}_i & \mathbf{b}_i \\ \mathbf{b}_i^* & c_i \end{pmatrix} \succeq \mathbf{0}\right\}.$$

Note that $(\mathrm{D}_m)$, also called Shor's relaxation, is an SDP [3], i.e., a problem involving the minimization of a linear function subject to LMIs.

The [illegible] [4] states that one always has $\mathrm{val}(\mathrm{D}_m) \le \mathrm{val}(\mathrm{QP}_m)$. A fundamental question is whether or not there is [illegible], i.e., is $\mathrm{val}(\mathrm{QP}_m) = \mathrm{val}(\mathrm{D}_m)$? When all the functions $f_i, i = 0,\dots,m$, are convex and strict feasibility holds, the answer is affirmative (this follows from the well-known strong duality theorem for convex programming [26]). However, if even one of the functions is not convex, then strong duality can be violated. Two exceptions are (i) the case of a single quadratic constraint ($m = 1$) (see, e.g., [10, 22]) and (ii) the case of two quadratic constraints ($m = 2$) in which the underlying number field is complex (strong duality is guaranteed when $\mathbb{F} = \mathbb{R}$). The latter result was recently derived in [1] and is recalled in Theorem 2.1.

Theorem 2.1 (see [1]). $\mathbb{F} = \mathbb{C}$ $(QP_2)$ i.e., $\tilde{\mathbf{z}} \in \mathbb{F}^n$ $f_1(\tilde{\mathbf{z}}) < 0, f_2(\tilde{\mathbf{z}}) < 0$

$$\exists \gamma_1 \geq 0, \gamma_2 \geq 0 : \gamma_1 \mathbf{A}_1 + \gamma_2 \mathbf{A}_2 \succ \mathbf{0}. \tag{9}$$

2 2 $(\ \ 2) = (\ \ 2)$

## 3. The two-quadratic Chebyshev center.

We now apply the results of the previous section to the problem of finding the Chebyshev center of the intersection of two level sets of quadratic functions. Specifically, we show that if the underlying number field is complex ($\mathbb{F} = \mathbb{C}$), then the Chebyshev center can be found by solving a convex optimization problem with two variables, or an SDP with three variables, thus rendering the problem tractable. In the case $\mathbb{F} = \mathbb{R}$ the proposed methodology results in an approximation of the exact Chebyshev center.

Consider the set $\Omega$ given as the intersection of level sets of two quadratic functions:

$$\Omega = \{\mathbf{z} \in \mathbb{F}^n : f_i(\mathbf{z}) \leq 0, i = 1, 2\}, \tag{10}$$

where $f_i(\mathbf{z}) = \mathbf{z}^* \mathbf{A}_i \mathbf{z} + 2\Re(\mathbf{b}_i^* \mathbf{z}) + c_i$ with $\mathbf{A}_i = \mathbf{A}_i^* \in \mathbb{F}^{n \times n}, \mathbf{b}_i \in \mathbb{F}^n$, and $c_i \in \mathbb{R}$ for $i = 1, 2$. We assume that condition (9) holds true. This is the case, for example, when at least one of the functions is strictly convex, which is equivalent to saying that the corresponding level set is a nondegenerate ellipsoid.

The Chebyshev center of $\Omega$ is the vector $\hat{\mathbf{z}} \in \mathbb{F}^n$ which is the solution to

$$\min_{\hat{\mathbf{z}} \in \mathbb{F}^n} \max_{\mathbf{z} \in \Omega} \|\mathbf{z} - \hat{\mathbf{z}}\|^2. \tag{11}$$

Theorem 3.1 below shows that finding the Chebyshev center of $\Omega$ can be recast as a convex optimization problem with only two variables. In order to prove the theorem, we will require the following lemma on Schur complements of singular matrices.

Lemma 3.1 (see [6, Appendix A.5]).

$$\mathbf{X} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{C} \end{pmatrix},$$

$\mathbf{A} = \mathbf{A}^* \in \mathbb{F}^{k \times k}, \mathbf{B} \in \mathbb{F}^{k \times p}$ $\mathbf{C} = \mathbf{C}^* \in \mathbb{F}^{p \times p}$ $\mathbf{X} \succeq \mathbf{0}$

$$\mathbf{A} \succeq \mathbf{0}, \quad \mathbf{C} - \mathbf{B}^* \mathbf{A}^\dagger \mathbf{B} \succeq \mathbf{0}, \quad (\mathbf{I} - \mathbf{A}\mathbf{A}^\dagger)\mathbf{B} = \mathbf{0}.$$

3.1. Note that the condition $(\mathbf{I} - \mathbf{A}\mathbf{A}^\dagger)\mathbf{B} = \mathbf{0}$ is equivalent to saying that $\mathbf{A}\mathbf{Y} = \mathbf{B}$ for some $\mathbf{Y} \in \mathbb{F}^{p \times k}$.

Theorem 3.1. $\Omega$ (10) $\mathbb{F} = \mathbb{C}$ $\tilde{\mathbf{z}} \in \mathbb{F}^n$ $f_1(\tilde{\mathbf{z}}) < 0$ $f_2(\tilde{\mathbf{z}}) < 0$ (9) (11)

$$\hat{\mathbf{z}} = -\left(\alpha_1 \mathbf{A}_1 + \alpha_2 \mathbf{A}_2\right)^{-1} \left(\alpha_1 \mathbf{b}_1 + \alpha_2 \mathbf{b}_2\right), \tag{12}$$

[illegible] $(\alpha_1, \alpha_2)$ [illegible]

$$\text{(13)}\quad \begin{array}{ll} \min_{\alpha_1,\alpha_2} & \left\{-c_1\alpha_1 - c_2\alpha_2 + (\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)^*(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)\right\} \\ & \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 \succeq \mathbf{I}, \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array}$$

[illegible]. Problem (11) can be rewritten as

$$\min_{\hat{\mathbf{z}} \in \mathbb{F}^n} \left\{ \|\hat{\mathbf{z}}\|^2 + \max_{\mathbf{z} \in \Omega} \left\{ \|\mathbf{z}\|^2 - 2\mathbf{z}^*\hat{\mathbf{z}} \right\} \right\}.$$

By using the strong duality result of Theorem 2.1 (note that all the conditions are satisfied), we conclude that the value of the inner maximization

$$\max_{\mathbf{z} \in \Omega} \{\|\mathbf{z}\|^2 - 2\mathbf{z}^*\hat{\mathbf{z}}\}$$

is equal to the value of the dual minimization problem (see section 2):

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2,\lambda} & \lambda \\ \text{s.t.} & \begin{pmatrix} -\mathbf{I} & \hat{\mathbf{z}} \\ \hat{\mathbf{z}}^* & \lambda \end{pmatrix} + \alpha_1 \begin{pmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{b}_1^* & c_1 \end{pmatrix} + \alpha_2 \begin{pmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^* & c_2 \end{pmatrix} \succeq \mathbf{0}, \\ & \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array}$$

Therefore, we can write (11) as

$$\text{(14)}\quad \begin{array}{ll} \min_{\alpha_1,\alpha_2,\hat{\mathbf{z}},\lambda} & \{\lambda + \|\hat{\mathbf{z}}\|^2\} \\ \text{s.t.} & \begin{pmatrix} -\mathbf{I} + \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 & \hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 \\ (\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)^* & \lambda + \alpha_1 c_1 + \alpha_2 c_2 \end{pmatrix} \succeq \mathbf{0}, \\ & \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array}$$

Using Lemma 3.1 and Remark 3.1, problem (14) is equivalent to

$$\text{(15)}\quad \begin{array}{ll} \min_{\alpha_1,\alpha_2,\hat{\mathbf{z}},\lambda} & \{\lambda + \|\hat{\mathbf{z}}\|^2\} \\ \text{s.t.} & \mathcal{B}_\alpha \succeq \mathbf{0}, \\ & \hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 \in \mathcal{R}(\mathcal{B}_\alpha), \\ & \lambda + \alpha_1 c_1 + \alpha_2 c_2 \geq (\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)\mathcal{B}_\alpha^\dagger(\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2), \\ & \alpha_1 \geq 0, \alpha_2 \geq 0, \end{array}$$

where we defined

$$\mathcal{B}_\alpha \equiv -\mathbf{I} + \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2.$$

Noting that at the optimum we will have equality in the third constraint of (15), our problem reduces to

$$\text{(16)}\quad \begin{array}{ll} \min_{\alpha_1,\alpha_2,\hat{\mathbf{z}}} & \left\{-\alpha_1 c_1 - \alpha_2 c_2 + (\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)\mathcal{B}_\alpha^\dagger(\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2) + \|\hat{\mathbf{z}}\|^2\right\} \\ \text{s.t.} & \mathcal{B}_\alpha \succeq \mathbf{0}, \\ & \hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 \in \mathcal{R}(\mathcal{B}_\alpha), \\ & \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array}$$

The constraint $\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 \in \mathcal{R}(\mathcal{B}_\alpha)$ is satisfied if and only if there exists $\mathbf{w} \in \mathbb{F}^n$ such that $\hat{\mathbf{z}} + \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 = \mathcal{B}_\alpha\mathbf{w}$. Using this observation combined with the identity $\mathcal{B}_\alpha\mathcal{B}_\alpha^\dagger\mathcal{B}_\alpha = \mathcal{B}_\alpha$, (16) becomes

$$\text{(17)}\quad \begin{array}{ll} \min_{\alpha_1,\alpha_2,\mathbf{w}} & \{-\alpha_1 c_1 - \alpha_2 c_2 + \mathbf{w}^*\mathcal{B}_\alpha\mathbf{w} + \| - \alpha_1\mathbf{b}_1 - \alpha_2\mathbf{b}_2 + \mathcal{B}_\alpha\mathbf{w}\|^2\} \\ \text{s.t.} & \mathcal{B}_\alpha \succeq \mathbf{0}, \\ & \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array}$$

Fixing $(\alpha_1, \alpha_2)$ and minimizing with respect to $\mathbf{w}$, we obtain that an optimal $\mathbf{w}$ is any vector satisfying

$$\mathcal{B}_\alpha(\mathbf{I} + \mathcal{B}_\alpha)\mathbf{w} = \mathcal{B}_\alpha(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2).$$

Choosing $\mathbf{w} = (\mathbf{I} + \mathcal{B}_\alpha)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)$ together with the identity

$$\mathcal{B}_\alpha(\mathbf{I} + \mathcal{B}_\alpha)^{-1} = \mathbf{I} - (\mathbf{I} + \mathcal{B}_\alpha)^{-1}$$

leads to the following form of (11):

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2} & \{-\alpha_1 c_1 - \alpha_2 c_2 + (\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)^*(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)\} \\ \text{s.t.} & \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 \succeq \mathbf{I}, \\ & \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array} \tag{18}$$

Since the objective in (18) is convex and the constraints are convex conic constraints, the problem (18) is convex. Finally,

$$\begin{aligned} \hat{\mathbf{z}} &= -\alpha_1\mathbf{b}_1 - \alpha_2\mathbf{b}_2 + \mathcal{B}_\alpha\mathbf{w} \\ &= (-\mathbf{I} + \mathcal{B}_\alpha(\mathbf{I} + \mathcal{B}_\alpha)^{-1})(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2) \\ &= -(\mathbf{I} + \mathcal{B}_\alpha)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2), \end{aligned}$$

completing the proof. ∎

An immediate consequence of Theorem 3.1 is that at the expense of adding an additional variable, we can recast the problem of finding the Chebyshev center of $\Omega$ as an SDP with three variables.

COROLLARY 3.2. [illegible] 3.1 [illegible] (11) [illegible]

$$\hat{\mathbf{z}} = -\left(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2\right)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2), \tag{19}$$

[illegible] $(\alpha_1, \alpha_2)$ [illegible]

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2,t} & \{-\alpha_1 c_1 - \alpha_2 c_2 + t\} \\ \text{s.t.} & \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 \succeq \mathbf{I}, \\ & \begin{pmatrix} \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 & \alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2 \\ (\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)^* & t \end{pmatrix} \succeq \mathbf{0}, \\ & \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array} \tag{20}$$

[illegible]. The proof follows from rewriting (13) as

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2,t} & \{-\alpha_1 c_1 - \alpha_2 c_2 + t\} \\ \text{s.t.} & \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 \succeq \mathbf{I}, \\ & (\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)^*(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2) \leq t, \\ & \alpha_1 \geq 0, \alpha_2 \geq 0, \end{array}$$

and invoking Lemma 3.1. ∎

Since problem (20) is an SDP, it can be solved efficiently via interior point methods [23]. Alternatively, we may solve the convex optimization problem (13) using the ellipsoid method [3], which is attractive given the small number of variables (two). In section 5 we compare these approaches.

The Chebyshev center of $\Omega$ can be calculated using Theorem 3.1 only when $\mathbb{F} = \mathbb{C}$. In the real case ($\mathbb{F} = \mathbb{R}$), strong duality is not guaranteed, and therefore the vector $\hat{\mathbf{z}}$

defined by (12), with $(\alpha_1, \alpha_2)$ being the optimal solution of (13) (or of (20)), is not necessarily the exact Chebyshev center. In fact, the weak duality theorem implies that the resulting ball will enclose the set but will not necessarily be the smallest one possible. In Figure 2, four examples of intersections of ellipsoids in the real domain are given. The vector $\hat{\mathbf{z}}$ was calculated by solving the SDP problem (20) with the software package SeDuMi [27]. The radius of each ball is the square root of the corresponding optimal value of problem (20). In the two upper examples it seems that strong duality holds while in the two lower examples it is evident that the circle defined by Theorem 3.1 (or Corollary 3.2) is not minimal.

An important property of an solution $(\bar{\alpha}_1, \bar{\alpha}_2)$ of problem (13) is that the matrix $\bar{\alpha}_1\mathbf{A}_1 + \bar{\alpha}_2\mathbf{A}_2 - \mathbf{I}$ is positive definite, i.e., the minimum eigenvalue of $\bar{\alpha}_1\mathbf{A}_1 + \bar{\alpha}_2\mathbf{A}_2 - \mathbf{I}$ is zero. This is proved in Theorem 3.3 below. This result is valid both in the complex and real domains. In section 4.2 we use this result in order to further reduce (13) to a convex optimization problem when $\mathbf{L} = \mathbf{I}$.

Theorem 3.3. $\tilde{\mathbf{z}} \in \mathbb{F}^n$ $f_1(\tilde{\mathbf{z}}) < 0$ $f_2(\tilde{\mathbf{z}}) < 0$ (9) $(\bar{\alpha}_1, \bar{\alpha}_2)$ (13) $\lambda_{\min}(\bar{\alpha}_1\mathbf{A}_1 + \bar{\alpha}_2\mathbf{A}_2 - \mathbf{I}) = 0$

Denote the objective function in (13) by

$$h(\boldsymbol{\alpha}) = -c_1\alpha_1 - c_2\alpha_2 + (\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2)^*(\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2)^{-1}(\alpha_1\mathbf{b}_1 + \alpha_2\mathbf{b}_2).$$

Then the following hold:

(i) $h(\boldsymbol{\alpha})$ is homogeneous, i.e., $h(\lambda\boldsymbol{\alpha}) = \lambda h(\boldsymbol{\alpha})$ for every $\lambda \neq 0$ and feasible $\boldsymbol{\alpha}$.

(ii) $h(\bar{\boldsymbol{\alpha}}) > 0$.

The first property is obvious by a simple substitution. To prove the second property note that by the weak duality theorem, $h(\bar{\boldsymbol{\alpha}})$ is greater than or equal to the value of the min-max problem (11). Let $\hat{\mathbf{z}}$ be the optimal solution of (11). Then $h(\bar{\boldsymbol{\alpha}}) \geq \max_{\mathbf{z}\in\Omega} \|\mathbf{z} - \hat{\mathbf{z}}\|^2$ and $\max_{\mathbf{z}\in\Omega} \|\mathbf{z} - \hat{\mathbf{z}}\|^2$ must be positive since, by our assumptions, $\Omega$ has a nonempty interior.

Suppose that $\bar{\alpha}_1\mathbf{A}_1 + \bar{\alpha}_2\mathbf{A}_2 \succ \mathbf{I}$. Then there exists $0 < \lambda < 1$ such that $\lambda\bar{\alpha}_1\mathbf{A}_1 + \lambda\bar{\alpha}_2\mathbf{A}_2 \succ \mathbf{I}$ so that $(\lambda\bar{\alpha}_1, \lambda\bar{\alpha}_2)$ is a feasible point of (13). However, from properties (i) and (ii),

$$h(\lambda\bar{\boldsymbol{\alpha}}) = \lambda h(\bar{\boldsymbol{\alpha}}) < h(\bar{\boldsymbol{\alpha}}),$$

contradicting the optimality of $\bar{\boldsymbol{\alpha}}$. □

## 4. The RCC estimator.

### 4.1. The RCC: Definition and form.

We now return to the problem of finding the Chebyshev center of FPS (5), which is the solution of the min-max problem (6). The set FPS can be represented as an intersection of two ellipsoids:

$$\text{FPS} = \{\mathbf{z} \in \mathbb{F}^n : \mathbf{z}^*\mathbf{L}^*\mathbf{L}\mathbf{z} \leq \eta, \quad \mathbf{z}^*\mathbf{A}^*\mathbf{A}\mathbf{z} - 2\Re(\mathbf{b}^*\mathbf{A}\mathbf{z}) + \|\mathbf{b}\|^2 \leq \rho\}.$$

We assume that condition (9) is satisfied, which means that

$$\exists\gamma_1 \geq 0, \gamma_2 \geq 0, \quad \gamma_1\mathbf{L}^*\mathbf{L} + \gamma_2\mathbf{A}^*\mathbf{A} \succ \mathbf{0}. \tag{21}$$

By Theorem 3.1, if $\mathbb{F} = \mathbb{C}$ and there exists $\tilde{\mathbf{z}}$ such that $\|\mathbf{A}\tilde{\mathbf{z}} - \mathbf{b}\|^2 < \rho$, $\|\mathbf{L}\tilde{\mathbf{z}}\|^2 < \eta$, then the Chebyshev center of FPS has the form

$$\hat{\mathbf{z}} = \alpha_2(\alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b},$$

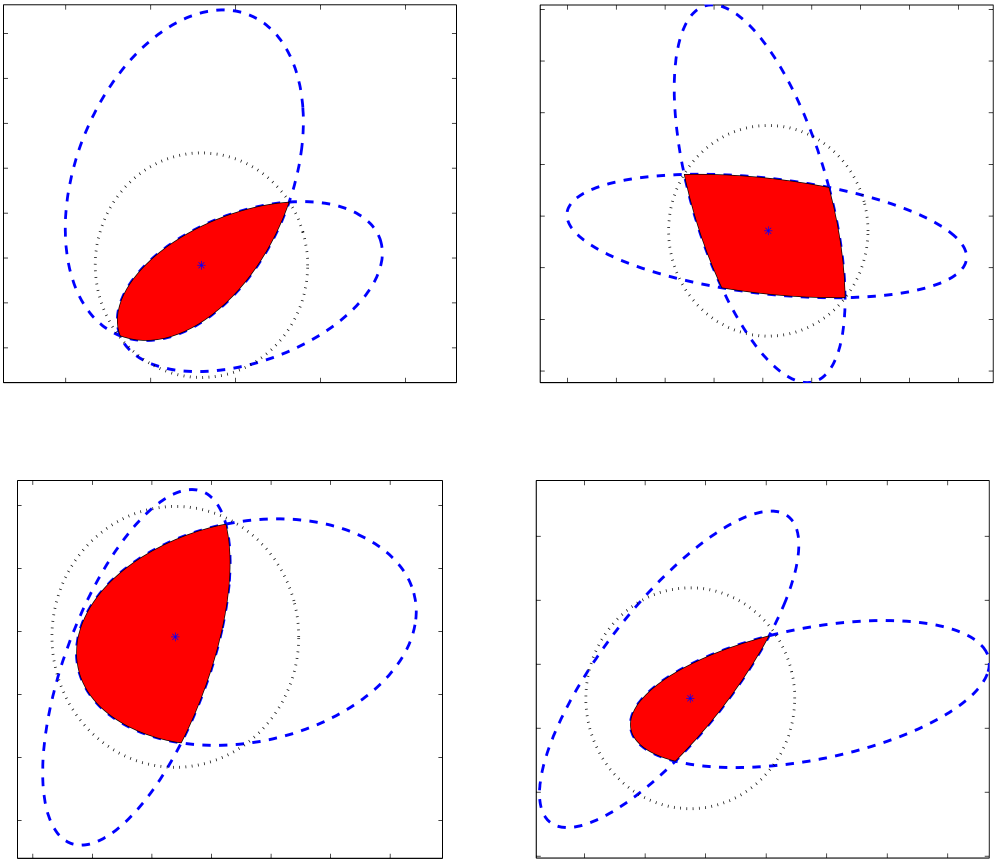


Fig. 2. *Four examples of intersection of ellipsoids (dashed lines). The filled area is the intersection of the ellipsoids. The center of the dotted circle is given by* (12) *with* $(\alpha_1, \alpha_2)$ *being an optimal solution of* (13) *and the radius being the square root of the corresponding optimal value.*

where $(\alpha_1, \alpha_2)$ is an optimal solution of the problem

$$\begin{array}{lll} & \min_{\alpha_1,\alpha_2} & \left\{\alpha_1\eta + \alpha_2(\rho - \|\mathbf{b}\|^2) + \alpha_2^2\mathbf{b}^*\mathbf{A}(\alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b}\right\} \\ (22) & \text{s.t.} & \alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}^*\mathbf{A} \succeq \mathbf{I}, \\ & & \alpha_1, \alpha_2 \geq 0. \end{array}$$

We now define $\hat{\mathbf{z}}$ for both the real and complex domains, and for the case when the conditions stated above are not necessarily satisfied.

Definition 4.1. [illegible]

$$\hat{\mathbf{z}}_{\mathrm{RCC}} = \alpha_2(\alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{b},$$

[illegible] $(\alpha_1, \alpha_2)$ [illegible] (22)

If the optimal $\alpha_2$ is positive, then the RCC estimator can be written as

$$\hat{\mathbf{z}}_{\mathrm{RCC}} = (\mathbf{A}^*\mathbf{A} + (\alpha_1/\alpha_2)\mathbf{L}^*\mathbf{L})^{-1}\mathbf{A}^*\mathbf{b}. \tag{23}$$

Therefore, the RCC estimator is essentially a Tikhonov regularization with a special choice of $\lambda$ that also takes into account the bounded noise constraint. This is in contrast to the choice of the regularization parameter in the RLS estimator that exploits only the norm constraint $\|\mathbf{Lz}\|^2 \leq \eta$.

In section 5 we demonstrate that although the RCC estimator is only an approximation of the Chebyshev center in the real domain, it can still significantly outperform the LS and RLS methods with respect to the estimation error. This is the case even when the bound on the noise is loose; thus, with almost the same information as used by the RLS approach, we can significantly reduce the estimation error by using our proposed strategy. The two key ingredients that lead to the improved performance are treating the estimation error directly and the added constraint on the noise.

**4.2. The case $\mathbf{L} = \mathbf{I}$.** We now show that in the interesting special case $\mathbf{L} = \mathbf{I}$, the task of calculating the RCC estimator reduces to a single-variable convex minimization problem. To this end we rely on Theorem 3.3.

THEOREM 4.1. $\mathbf{L} = \mathbf{I}$ $\delta = \lambda_{\min}(\mathbf{A}^*\mathbf{A})$

$$\hat{\mathbf{z}}_{\mathrm{RCC}} = \begin{cases} (\mathbf{A}^*\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}, & 0 \leq \lambda < \infty, \\ \mathbf{0}, & \lambda = \infty, \end{cases}$$

$\lambda$ [3]

(i) $\delta > 0$ $\lambda = 1/\mu - \delta$ $\mu$

$$\min_{0 \leq \mu \leq 1/\delta} \left\{(1-\delta\mu)\eta + \mu(\rho - \|\mathbf{b}\|^2) + \mu^2\mathbf{b}^*\mathbf{A}(\mu(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}\right\}. \tag{24}$$

(ii) $\delta = 0$ $\lambda = 1/\xi$ $\xi$

$$\min_{\xi \geq 0} \left\{\xi(\rho - \|\mathbf{b}\|^2) + \xi^2\mathbf{b}^*\mathbf{A}(\xi\mathbf{A}^*\mathbf{A} + \mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}\right\}. \tag{25}$$

Substituting $\mathbf{L} = \mathbf{I}$ into (22) we find

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2} & \left\{\alpha_1\eta + \alpha_2(\rho - \|\mathbf{b}\|^2) + \alpha_2^2\mathbf{b}^*\mathbf{A}(\alpha_2\mathbf{A}^*\mathbf{A} + \alpha_1\mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}\right\} \\ \text{s.t.} & \alpha_2\mathbf{A}^*\mathbf{A} + \alpha_1\mathbf{I} \succeq \mathbf{I}, \\ & \alpha_1, \alpha_2 \geq 0. \end{array} \tag{26}$$

The LMI constraint can be written equivalently as

$$\alpha_2\lambda_{\min}(\mathbf{A}^*\mathbf{A}) + \alpha_1 = \alpha_2\delta + \alpha_1 \geq 1. \tag{27}$$

From Theorem 3.3, we conclude that (27) must be satisfied with equality. Therefore,

$$\alpha_1 = 1 - \delta\alpha_2. \tag{28}$$

Substituting (28) into (26) we obtain that in the case $\delta > 0$, (26) becomes

$$\begin{array}{ll} \min_{\alpha_2} & \left\{(1-\delta\alpha_2)\eta + \alpha_2(\rho - \|\mathbf{b}\|^2) + \alpha_2^2\mathbf{b}^*\mathbf{A}(\alpha_2(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}\right\} \\ \text{s.t.} & 0 \leq \alpha_2 \leq 1/\delta, \end{array}$$

---

[3] We use the standard terminology $\frac{a}{0} = \infty$ whenever $a > 0$.

which is the same as (24) after $\mu$ is replaced by $\alpha_2$. The result for the case $\delta = 0$ is similarly derived. $\square$

To solve the single-variable convex problems (24) and (25), we can use any solver of one-dimensional convex minimization problems—for instance, a simple bisection algorithm on the derivative of the function. Denoting by $q(\mu)$ and $q'(\mu)$ the objective in (24) and its derivative, respectively, we have

$$\begin{aligned} q'(\mu) = {} & -\delta\eta + \rho - \|\mathbf{b}\|^2 + 2\mu\mathbf{b}^*\mathbf{A}(\mu(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I})^{-1}\mathbf{A}^*\mathbf{b} \\ & -\mu^2\mathbf{b}^*\mathbf{A}(\mu(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I})^{-1}(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I})(\mu(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}. \end{aligned}$$

Since $\mu(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I}$ is a positive definite matrix for every choice of $\mu \geq 0$, we can calculate the derivative using a single Cholesky factorization in the following manner.

[illegible] $q'(\mu)$.

```
1. Calculate a Cholesky factorization $\mathbf{D}^*\mathbf{D} = \mu(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I}) + \mathbf{I}$.
2. Solve the system $\mathbf{D}^*\mathbf{y} = \mathbf{A}^*\mathbf{b}$.
3. Solve the system $\mathbf{D}\mathbf{x} = \mathbf{y}$.
4. The derivative is given by $q'(\mu) = -\delta\eta + \rho - \|\mathbf{b}\|^2 + 2\mu\mathbf{b}^*\mathbf{A}\mathbf{x} - \mu^2\mathbf{x}^*(\mathbf{A}^*\mathbf{A} - \delta\mathbf{I})\mathbf{x}$.
```

Note that the Cholesky factorization is the most expensive component in the calculation of $q'(\mu)$ (the calculation of $\mathbf{A}^*\mathbf{A}$ is done in a preprocess). The other operations—solution of triangular systems and matrix/vector multiplications—are significantly cheaper. An alternative approach for computing the derivative is using the singular value decomposition of $\mathbf{A}$. This approach is viable for small-size problems but is not applicable for medium- and large-scale problems in which the Cholesky or the sparse Cholesky factorization can be employed. The complete description of the algorithm for calculating the RCC estimator when $\mathbf{L} = \mathbf{I}$ and $\delta > 0$ is as follows.

**Algorithm RCC-S.**

**Input:** $\mathbf{A} \in \mathbb{F}^{m\times n}$, the model matrix; $\mathbf{b} \in \mathbb{F}^m$, the (noisy) right-hand side vector; $\eta$, an upper bound on $\|\mathbf{z}\|^2$; and $\rho$, an upper bound on the squared-norm of the noise $\|\mathbf{A}\mathbf{z} - \mathbf{b}\|^2$.

**Output:** The RCC estimator $\hat{\mathbf{z}}_{\mathrm{RCC}}$, which is the solution to problem (22) with $\mathbf{L} = \mathbf{I}$.

```
1. If $q'(0) \geq 0$ then $h = 0$, and go to step 5.
2. If $q'(1/\delta) \leq 0$ then $h = 1/\delta$, and go to step 5.
3. Set $lb = 0, ub = 1/\delta$.
4. Repeat the following steps until $|ub - lb| < \eta$:
    (a) Set $h = \frac{lb+ub}{2}$.
    (b) Calculate $d = q'(h)$.
    (c) If $d \geq 0$ then $ub = h$; else $lb = h$.
5. Set $\hat{\mathbf{z}}_{\mathrm{RCC}} = (\mathbf{A}^*\mathbf{A} + (1 - \delta h)/h\mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}$.
```

A similar algorithm can be defined for the case $\delta = 0$.

**5. Numerical examples.** We now present some examples comparing the RCC estimator with the LS and RLS methods, given by (2) and (3), respectively. The comparison was employed on two sets of problems: randomly chosen problems and the discretized inverse heat equation from "Regularization tools" [15]. All experiments were performed in MATLAB.

We note that in the simulations we assume knowledge of a very loose bound $\rho$ on the noise so that essentially our method exploits almost the same knowledge as the RLS approach.

**5.1. Random test problems.** We chose a problem with dimensions $m = 10, n = 7$. Each component of $\mathbf{A}$ was randomly generated from a uniform distribution (i.e., $\mathbf{A} = \text{rand}(m, n)$). The "true" vector $\mathbf{z}_{\text{T}}$ is the vector of all ones. In the constraints, $\mathbf{L} = \mathbf{I}$ and $\eta = 2\|\mathbf{z}_{\text{T}}\|^2$. The observed vector $\mathbf{b}$ was generated by

$$\mathbf{b} = \mathbf{A}\mathbf{z}_{\text{T}} + \sigma\mathbf{w},$$

where $\sigma$ takes the values $0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$ and each component of $\mathbf{w}$ was randomly generated from a standard normal distribution. The upper bound on the squared norm of the noise was chosen as $10\|\mathbf{w}\|^2$ (i.e., 10 times the true squared norm).

Table 1 describes the average of the data error $\|\mathbf{A}\hat{\mathbf{z}} - \mathbf{b}\|^2$ (here $\hat{\mathbf{z}}$ is $\hat{\mathbf{z}}_{\text{LS}}, \hat{\mathbf{z}}_{\text{RLS}}$, or $\hat{\mathbf{z}}_{\text{RCC}}$) and the squared error residual $\|\mathbf{z}_{\text{T}} - \hat{\mathbf{z}}\|^2$ over 100 realizations of $\mathbf{w}$. The best results in each half row are marked in boldface. The RLS approach (3) was implemented using the function lsqi from [15] and the RCC estimator was generated by the RCC-S algorithm of section 4.2.

TABLE 1

*Comparison of the LS, RLS, and RCC estimators with respect to estimation error and data error.*

| $\sigma$ | Squared estimation error | | | Squared data error | | |
|---|---|---|---|---|---|---|
| | LS | RLS | RCC | LS | RLS | RCC |
| 0.01 | **1.9e-3** | **1.9e-3** | **1.9e-3** | **3.0e-4** | **3.0e-4** | **3.0e-4** |
| 0.1 | 2.1e-1 | 2.1e-1 | **1.8e-1** | **3.0e-2** | **3.0e-2** | 3.1e-2 |
| 0.2 | 6.6e-1 | 6.6e-1 | **3.6e-1** | **1.2e-1** | **1.2e-1** | 1.5e-1 |
| 0.3 | 1.8e+0 | 1.8e+0 | **2.0e-1** | **2.6e-1** | **2.6e-1** | 1.2e+0 |
| 0.4 | 3.1e+0 | 2.9e+0 | **2.5e-1** | **5.3e-1** | **5.3e-1** | 2.7e+0 |
| 0.5 | 4.3e+0 | 3.9e+0 | **3.3e-1** | **7.2e-1** | 7.3e-1 | 4.8e+0 |
| 0.6 | 6.5e+0 | 5.0e+0 | **4.7e-1** | **1.1e+0** | 1.2e+0 | 7.3e+0 |
| 0.7 | 9.6e+0 | 5.4e+0 | **5.5e-1** | **1.5e+0** | 1.7e+0 | 1.0e+1 |
| 0.8 | 1.4e+1 | 6.6e+0 | **6.8e-1** | **2.1e+0** | 2.4e+0 | 1.4e+1 |
| 0.9 | 1.5e+1 | 6.7e+0 | **9.0e-1** | **2.4e+0** | 2.9e+0 | 1.8e+1 |
| 1.0 | 1.8e+1 | 6.8e+0 | **9.9e-1** | **2.8e+0** | 3.6e+0 | 2.2e+1 |

It is evident that the LS and RLS estimators are significantly and consistently worse than the RCC method with respect to the estimation error. This is despite the fact that the bound on the data error was chosen to be very large—much larger than the true bound. Thus, this approach does not require much prior information. On the other hand, the LS and RLS estimators result in a smaller data error than the RCC approach. This is not surprising since, as was already mentioned, the RCC method is designed to minimize a measure of estimation error while the LS and RLS strategies are aimed at minimizing the data error, which is less relevant in an [illegible] context.

Note that the RCC method was implemented in the case $\mathbb{F} = \mathbb{R}$. Recall that in the real domain the RCC estimator is only an approximation of the Chebyshev center of the FPS. We also implemented a set of random examples over $\mathbb{F} = \mathbb{C}$. The results were essentially the same as those reported in Table 1. Therefore, it seems that at least from an empirical point of view the RCC strategy is a "good enough" approximation of the Chebyshev center.

As we already pointed out, both the RCC and RLS strategies are Tikhonov estimators with different regularization parameters. In all the simulations in this section we observed that the regularization parameter of the RCC method,[4] denoted by $\lambda_{\text{RCC}}$,

---

[4] $\alpha_2$ was always nonzero in our experiments.

is consistently greater than or equal to the parameter $\lambda_{\mathrm{RLS}}$ of the RLS approach. Furthermore, the RCC estimator was always feasible so that $\|\mathbf{L}\hat{\mathbf{z}}_{\mathrm{RCC}}\|^2 \leq \eta$. The latter observation explains why $\lambda_{\mathrm{RCC}} \geq \lambda_{\mathrm{RLS}}$. To see this, we define $\varphi(\lambda) \equiv \|\mathbf{L}\mathbf{z}_\lambda\|^2$, where $\mathbf{z}_\lambda$ is given by (4). It is straightforward to show that the function $\varphi$ is strictly decreasing under our assumption (21). Now, if $\|\mathbf{L}\hat{\mathbf{z}}_{\mathrm{LS}}\|^2 \leq \eta$, then $\lambda_{\mathrm{RLS}} = 0$, which immediately implies $\lambda_{\mathrm{RCC}} \geq \lambda_{\mathrm{RLS}}$. Otherwise, when $\|\mathbf{L}\hat{\mathbf{z}}_{\mathrm{LS}}\|^2 > \eta$, $\lambda_{\mathrm{RLS}}$ satisfies $\varphi(\lambda_{\mathrm{RLS}}) = \eta$. On the other hand, $\varphi(\lambda_{\mathrm{RCC}}) = \|\mathbf{L}\hat{\mathbf{z}}_{\mathrm{RCC}}\|^2 \leq \eta$, and by the fact that $\varphi$ is decreasing, $\lambda_{\mathrm{RCC}} \geq \lambda_{\mathrm{RLS}}$.

**5.2. Inverse heat equation.** We now treat the problem of estimating the function $f(t)$ that solves the heat equation

$$\int_0^1 k(s-t)f(t) = g(s),$$

with $k(t) = \frac{t^{-3/2}}{2\sqrt{\pi}}\exp\left(-\frac{1}{4t}\right)$. By means of a simple collocation and midpoint rule with $n$ points, the problem reduces to an $n \times n$ linear system $\mathbf{A}\mathbf{z}_{\mathrm{T}} = \mathbf{b}_{\mathrm{T}}$. This system and its solution $\mathbf{z}_{\mathrm{T}}$ are implemented in the function heat(n,1) from [15]. We note that this example is ill-conditioned. We compare the RCC estimator to the RLS method (the results for the LS approach are not given because it produces extremely poor results).

The perturbed right-hand side is chosen as

$$\mathbf{b} = \mathbf{b}_{\mathrm{T}} + 10^{-4}\mathbf{w}, \tag{29}$$

where each component of $\mathbf{w}$ is generated from a standard normal distribution. The matrix $\mathbf{L}$ approximates the first-derivative operator implemented in the function get_l(n,1) from [15]. The upper bound $\eta$ was chosen to be $2\|\mathbf{L}\mathbf{z}_{\mathrm{T}}\|^2$. In Figure 3 three possible values of $\rho$ were employed: $\rho = \|\mathbf{w}\|^2$ (exact squared norm), $\rho = 2\|\mathbf{w}\|^2$, and $\rho = 10\|\mathbf{w}\|^2$. The results of the RCC estimator in these three cases are very similar and are much closer to the true vector $\mathbf{z}_{\mathrm{T}}$ than the RLS solution $\hat{\mathbf{z}}_{\mathrm{RLS}}$. Therefore, it seems that at least in this example, the performance of the RCC method is quite robust with respect to the choice of $\rho$. The fourth plot in Figure 3 describes the three vectors $\mathbf{A}\mathbf{z}_{\mathrm{T}}, \mathbf{A}\hat{\mathbf{z}}_{\mathrm{RLS}}$, and $\mathbf{A}\hat{\mathbf{z}}_{\mathrm{RCC}}$ (for $\rho = 10\|\mathbf{w}\|^2$). It can be readily seen that the three vectors are almost identical, implying that the data errors of the RLS and RCC approaches are both negligible.

**5.3. Ellipsoid versus interior-point methods.** In the case when $\mathbf{L} \neq \mathbf{I}$ (as in the inverse heat equation problem), we are required to solve the convex optimization problem (22) with two variables, or the SDP

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2,t} & \{\alpha_1\eta + \alpha_2(\rho - \|\mathbf{b}\|^2) + t\} \\ \text{s.t.} & \alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}^*\mathbf{A} \succeq \mathbf{I}, \\ & \begin{pmatrix} \alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}\mathbf{A} & -\alpha_2\mathbf{A}^*\mathbf{b} \\ -\alpha_2\mathbf{A}\mathbf{b}^* & t \end{pmatrix} \succeq \mathbf{0}, \alpha_1 \geq 0, \alpha_2 \geq 0. \end{array} \tag{30}$$

Now, consider an SDP of the general form

$$\min\left\{\mathbf{c}^T\mathbf{x} : \sum_{i=1}^m x_i\mathbf{B}_i \succeq \mathbf{E}\right\},$$

where $\mathbf{c} \in \mathbb{R}^m$ and $\mathbf{E}, \mathbf{B}_i, i = 1, \ldots, m$, are $n \times n$ Hermitian matrices. In order to solve the general form SDP we can use a primal-dual interior-point method, which

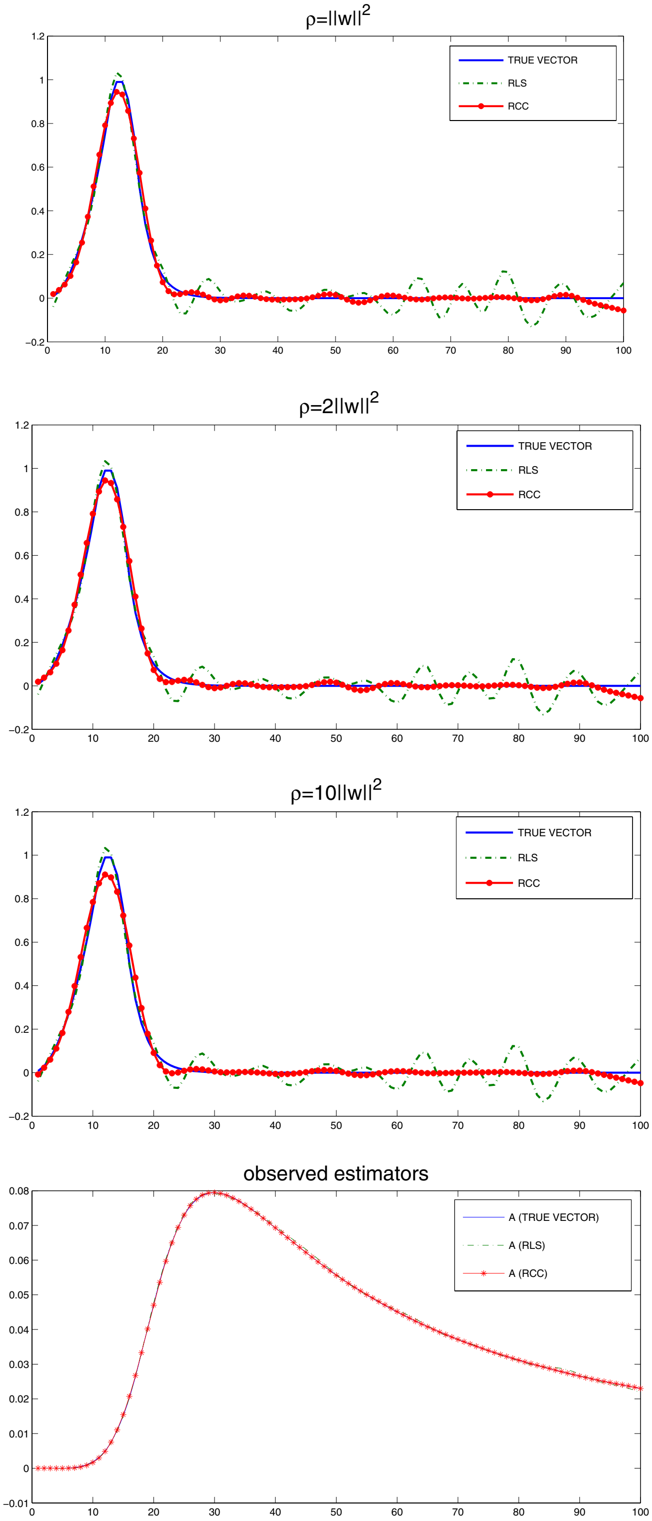


Fig. 3. *Results for the inverse heat problem of the RCC and RLS estimators.*

requires $O(n^{3.5}m^{1.5}+n^{2.5}m^2+n^{0.5}m^{2.5})$ operations per accuracy digit. For the specific problem (30) we have $m=3$, and the amount of operations is therefore $O(n^{3.5})$.

Another alternative is to use the ellipsoid method [3] directly on the problem (22). This algorithm requires $O(n^3)$ operations per accuracy digit since it requires at most two Cholesky factorizations at each iteration. Therefore, it is cheaper than the SDP approach by a factor of order $\sqrt{n}$. To compare the performance of the two algorithms, we implemented the ellipsoid method (see the appendix for full details) and compared it to the interior-point method implemented in SeDuMi [27] on the inverse heat equation problem with various values of $n$. The CPU time in seconds of the ellipsoid and interior-point algorithms averaged over 10 realizations of the noise $\mathbf{w}$ is given in Table 2 below ($\sigma$ was fixed to be 1e-4). For $n=1000$ SeDuMi failed due to memory difficulties. Table 2 demonstrates the efficiency of the ellipsoid method.

TABLE 2

*CPU time in seconds on a Pentium* 4, 1.8*Ghz.*

| $n$ | Ellipsoid | SeDuMi |
|---|---|---|
| 10 | 1.4e-1 | 5.5e-1 |
| 20 | 1.6e-1 | 9.3e-1 |
| 50 | 2.9e-1 | 3.5e+0 |
| 100 | 8.5e-1 | 1.8e+1 |
| 200 | 6.0e+0 | 1.0e+2 |
| 500 | 3.8e+1 | 8.3e+2 |
| 1000 | 2.4e+2 | – |

## 6. Extensions to other estimation problems.

The RCC estimator was constructed to handle the situation in which only the right-hand side of the linear system $\mathbf{Ax}\approx\mathbf{b}$ is contaminated by noise. The same methodology can be applied to deal with other sources of noise. In this section, we briefly outline the resulting estimators in two scenarios: (i) both $\mathbf{A}$ and $\mathbf{b}$ are uncertain, and (ii) $\mathbf{A}$ and $\mathbf{b}$ are uncertain and regularization is required. In the first scenario, the proposed estimator has a similar structure to the well-known TLS method [13, 17], and in the second scenario, the estimator has a form similar to that of the RTLS solution [12]. Thus, these popular methods of handling uncertainties in the basic regression model (1) can be shown to be special cases of our general results.

The derivation of the estimators is very similar to that described in section 3; therefore, we present the main results without proof.

### 6.1. Uncertainty in both A and b.

Suppose that both $\mathbf{A}$ and $\mathbf{b}$ are uncertain and are given by $\mathbf{A}+\boldsymbol{\Delta},\mathbf{b}+\mathbf{w}$ with $\boldsymbol{\Delta},\mathbf{w}$ being unknown but bounded perturbations. This setting is assumed in the robust LS approach [11]. We assume that the bound constraint is given by[5] $\|(\boldsymbol{\Delta},\mathbf{w})\|_{\mathrm{F}}^2\le\rho$. The corresponding FPS is

$$\mathrm{FPS}_1=\{\mathbf{z}\in\mathbb{F}^n:\exists\boldsymbol{\Delta}\in\mathbb{F}^{m\times n},\mathbf{w}\in\mathbb{F}^m:(\mathbf{A}+\boldsymbol{\Delta})\mathbf{z}=\mathbf{b}+\mathbf{w},\|(\boldsymbol{\Delta},\mathbf{w})\|_{\mathrm{F}}^2\le\rho\}.$$

To apply our results, we first note that $\mathrm{FPS}_1$ can be written as the single quadratic constraint

$$\mathrm{FPS}_1=\{\mathbf{z}\in\mathbb{F}^n:\mathbf{z}^*(\mathbf{A}^*\mathbf{A}-\rho\mathbf{I})\mathbf{z}-2\mathbf{z}^*\mathbf{A}^*\mathbf{b}+\|\mathbf{b}\|^2-\rho\le 0\}. \tag{31}$$

This follows from writing $\mathrm{FPS}_1$ as

$$\mathrm{FPS}_1=\{\mathbf{z}\in\mathbb{F}^n:\mathbf{Az}-\mathbf{b}=\mathbf{E}\tilde{\mathbf{z}}\text{ for some }\|\mathbf{E}\|_{\mathrm{F}}^2\le\rho\}$$

---

[5] For a matrix $\mathbf{B}$, $\|\mathbf{B}\|_{\mathrm{F}}$ denotes the Frobenius norm of $\mathbf{B}$.

and applying the following simple lemma.

LEMMA 6.1. [illegible] $\mathbf{x} \in \mathbb{F}^n$ [illegible] $\mathbf{y} \in \mathbb{F}^m$ [illegible] $\eta$ [illegible]

(i) [illegible] $\Delta \in \mathbb{F}^{m \times n}$ [illegible] $\Delta\mathbf{x} = \mathbf{y}$ [illegible] $\|\Delta\|_F \leq \eta$

(ii) [illegible] $\|\mathbf{y}\| \leq \eta\|\mathbf{x}\|$ [illegible]

Under the assumption that $\rho < \lambda_{\min}(\mathbf{A}^*\mathbf{A})$, it can be shown, using the same line of analysis of section 3, that the Chebyshev center of $\text{FPS}_1$ is given by

$$\hat{\mathbf{z}} = (\mathbf{A}^*\mathbf{A} - \rho\mathbf{I})^{-1}\mathbf{A}^*\mathbf{b}, \tag{32}$$

which is a [illegible] of the LS solution. Since $\text{FPS}_1$ consists of a single quadratic constraint, this result is valid both in the real and in the complex domains. We note that when $\rho > \lambda_{\min}(\mathbf{A}^*\mathbf{A})$, $\text{FPS}_1$ is unbounded, and as a result the value of the inner maximization problem in (6) is always $\infty$, which implies that the Chebyshev center in this case is meaningless.

If we choose

$$\rho = \lambda_{\min}\begin{pmatrix} \mathbf{A}^*\mathbf{A} & \mathbf{A}^*\mathbf{b} \\ \mathbf{b}^*\mathbf{A} & \|\mathbf{b}\|^2 \end{pmatrix},$$

then the estimator (32) coincides with the TLS estimator [17, Theorem 2.7].

### 6.2. Uncertainty in both A and b with regularization.

Suppose now we add regularization to the previous scenario; i.e., we consider the feasible set

$$\text{FPS}_2 = \{\mathbf{z} \in \mathbb{F}^n : \|\mathbf{L}\mathbf{z}\|^2 \leq \eta, \exists \mathbf{\Delta} \in \mathbb{F}^{m\times n}, \mathbf{w} \in \mathbb{F}^m : (\mathbf{A}+\mathbf{\Delta})\mathbf{z} = \mathbf{b}+\mathbf{w}, \|(\mathbf{\Delta}, \mathbf{w})\|_{\text{F}}^2 \leq \rho\},$$

which can also be written as

$$\text{FPS}_2 = \{\mathbf{z} \in \mathbb{F}^n : \|\mathbf{L}\mathbf{z}\|^2 \leq \eta, \mathbf{z}^*(\mathbf{A}^*\mathbf{A} - \rho\mathbf{I})\mathbf{z} - 2\mathbf{z}^*\mathbf{A}^*\mathbf{b} + \rho - \|\mathbf{b}\|^2 \leq 0\}.$$

In the case $\mathbb{F} = \mathbb{C}$, the Chebyshev center of $\text{FPS}_2$ is given by

$$\hat{\mathbf{z}} = \alpha_2(\alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2(\mathbf{A}^*\mathbf{A} - \rho\mathbf{I}))^{-1}\mathbf{A}^*\mathbf{b}, \tag{33}$$

where $(\alpha_1, \alpha_2)$ is an optimal solution of the convex optimization problem

$$\begin{array}{ll} \min_{\alpha_1,\alpha_2} & \{\alpha_1\eta + \alpha_2(\rho - \|\mathbf{b}\|^2) + \alpha_2^2\mathbf{b}^*\mathbf{A}(\alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2(\mathbf{A}^*\mathbf{A} - \rho\mathbf{I}))^{-1}\mathbf{A}^*\mathbf{b}\} \\ \text{s.t.} & \alpha_1\mathbf{L}^*\mathbf{L} + \alpha_2\mathbf{A}^*\mathbf{A} \succeq \mathbf{I}, \\ & \alpha_1, \alpha_2 \geq 0. \end{array}$$

If $\alpha_2 \neq 0$ then $\hat{\mathbf{z}}$ of (33) can be written as

$$\hat{\mathbf{z}} = (\mathbf{A}^*\mathbf{A} - \rho\mathbf{I} + \alpha_1/\alpha_2\mathbf{L}^*\mathbf{L})^{-1}\mathbf{A}^*\mathbf{b}.$$

This estimator has the same structure as the RTLS method, which solves the equation

$$(\mathbf{A}^*\mathbf{b} - \lambda\mathbf{I} + \mu\mathbf{L}^*\mathbf{L})\mathbf{x}_{\text{RTLS}} = \mathbf{A}^*\mathbf{b}$$

for some choice of parameters $\lambda, \mu$ [12].

**7. Conclusion.** In this paper we discussed a Chebyshev center regularization method that is based on an estimation error criterion. In contrast to previous regularization strategies that invoke a data error–based criterion, here we focus on the estimation error and try to minimize it in some sense. Since the estimation error depends on the unknown vector, we choose as our estimate the Chebyshev center of an FPS, which consists of a constraint both on the data error and on the weighted norm of the true parameter. Although the resulting problem is nonconvex, by exploiting recent duality results, we show that in the complex domain it can be formulated as a solution to a convex optimization problem in two unknowns, and in the real case the same approach can be used to get a "pretty good" approximation of the true Chebyshev center. From a numerical standpoint, we provide two solution methods and compare their performance. The first is based on an SDP and the second on an ellipsoid algorithm. The latter turns out to be more efficient as the problem size grows. Finally, we show that the popular TLS and RTLS methods can also be formulated within our framework.

## Appendix. The ellipsoid method for problem (13).

In this appendix we describe in detail the ellipsoid method as applied to the convex optimization problem (13).

The two basic ingredients in the ellipsoid method are a separation oracle and a first-order oracle (see, e.g., [3]). The main linear algebra procedure we use in both oracles is the Cholesky factorization. We assume that the input to the Cholesky procedure is a symmetric matrix $\mathbf{B}$, and its output consists of three arguments **flag,** $\mathbf{D}$, and $\mathbf{x}$. If $\mathbf{flag} = \mathbf{1}$ then $\mathbf{B}$ is positive definite, $\mathbf{B} = \mathbf{D}^*\mathbf{D}$ with $\mathbf{D}$ being a lower triangular matrix, and $\mathbf{x}$ is NULL. If $\mathbf{flag} = \mathbf{0}$ then $\mathbf{B}$ is not positive definite, $\mathbf{x}$ is a vector satisfying $\mathbf{x}^*\mathbf{B}\mathbf{x} \leq 0$, and $\mathbf{D}$ is NULL.

The input to the separation oracle is a vector $\boldsymbol{\alpha} \in \mathbb{R}^2$. The output is either a statement that the vector is feasible (up to some tolerance) or a hyperplane separating the vector from the feasible set. $\epsilon$ is a tolerance parameter chosen as $10^{-6}$ in our implementation.

```
Algorithm SEP-ORA.
Input: $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T \in \mathbb{R}^2$.
Output: flag equals one if $\boldsymbol{\alpha}$ is feasible (up to some tolerance) and zero otherwise.
$\mathbf{d} \in \mathbb{R}^2$ is a separating hyperplane.
    1. If $\alpha_1 \leq -\epsilon$ then flag=0, $\mathbf{d} = (-1, 0)^T$, STOP.
    2. If $\alpha_2 \leq -\epsilon$ then flag=0, $\mathbf{d} = (0, -1)^T$, STOP.
    3. Set $\mathbf{M} = \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2 - \mathbf{I} + \epsilon\mathbf{I}$.
    4. Invoke the Cholesky factorization procedure with input $\mathbf{M}$ and obtain an
       output {flag, D, x}.
        (a) If flag = 1 then STOP.
        (b) If flag = 0 then $(d_1, d_2) = (\mathbf{x}^*\mathbf{A}_1\mathbf{x}, \mathbf{x}^*\mathbf{A}_2\mathbf{x})$, STOP.
```

The first-order oracle is invoked in the case when the current vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ is feasible. Its main computational effort is the Cholesky factorization of the matrix $\alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2$, which by feasibility of $\boldsymbol{\alpha}$, must be positive definite.

```
Algorithm FO-ORA.
Input: $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T \in \mathbb{R}^2$, an $\eta$-feasible solution of (13).
Output: f, the gradient of the objective function of (13) at $\boldsymbol{\alpha}$.
    1. Set $\mathbf{M} = \alpha_1\mathbf{A}_1 + \alpha_2\mathbf{A}_2$.
    2. Invoke the Cholesky factorization procedure with input $\mathbf{M}$ and obtain an
       output $\{\mathbf{flag}, \mathbf{D}, \mathbf{x}\}$.
```

```
3. Solve the following linear systems in $\mathbf{x}_1, \mathbf{x}_2$: $\mathbf{D}^*\mathbf{x}_1 = \mathbf{b}_1, \mathbf{D}^*\mathbf{x}_2 = \mathbf{b}_2$.
4. Solve the following linear systems in $\mathbf{v}_1, \mathbf{v}_2$: $\mathbf{D}\mathbf{v}_1 = \mathbf{x}_1, \mathbf{D}\mathbf{v}_2 = \mathbf{x}_2$.
5. Set
     $f_1 = -c_1 + 2\alpha_1 \mathbf{b}_1^* \mathbf{v}_1 - \alpha_1^2 \mathbf{v}_1^* \mathbf{A}_1 \mathbf{v}_1 + 2\alpha_2 \mathbf{b}_1^* \mathbf{v}_2 - 2\alpha_1\alpha_2 \mathbf{v}_1^* \mathbf{A}_1 \mathbf{v}_2 - \alpha_2^2 \mathbf{v}_2^* \mathbf{A}_1 \mathbf{v}_2,$
     $f_2 = -c_2 + 2\alpha_2 \mathbf{b}_2^* \mathbf{v}_2 - \alpha_2^2 \mathbf{v}_2^* \mathbf{A}_2 \mathbf{v}_2 + 2\alpha_1 \mathbf{b}_1^* \mathbf{v}_2 - 2\alpha_1\alpha_2 \mathbf{v}_1^* \mathbf{A}_2 \mathbf{v}_2 - \alpha_1^2 \mathbf{v}_1^* \mathbf{A}_2 \mathbf{v}_1.$
```

We are now ready to describe the implementation of the ellipsoid method on the convex optimization problem (13).

**Algorithm Ellipsoid**.

**Input:** The optimization problem (13).

**Output:** $\boldsymbol{\alpha} \in \mathbb{R}^2$, a solution to problem (13) (up to some tolerance).

```
1. Set $R = 10^8, \mathbf{B} = R\mathbf{I}_2, \boldsymbol{\alpha} = (0,0)^T, v = \pi 10^{16}$.
2. Repeat the following steps until $v < \epsilon$.
   (a) Invoke the separation oracle SEP-ORA with input $\boldsymbol{\alpha}$ and obtain an output $\{\mathbf{flag}, \mathbf{d}\}$. If $\mathbf{flag} = \mathbf{0}$ then go to step (c).
   (b) Invoke the first-order oracle FO-ORA with input $\boldsymbol{\alpha}$ and obtain an output $\mathbf{d}$.
   (c) $\mathbf{p} = \frac{\mathbf{B}^T\mathbf{d}}{\sqrt{\mathbf{d}^T\mathbf{B}\mathbf{B}^T\mathbf{d}}}$.
   (d) $\boldsymbol{\alpha} = \boldsymbol{\alpha} - \frac{1}{3}\mathbf{B}\mathbf{p}$.
   (e) $\mathbf{B} = \frac{2}{\sqrt{3}}\mathbf{B} + (\frac{2}{3} - \frac{2}{\sqrt{3}})\mathbf{B}\mathbf{p}\mathbf{p}^T$.
   (f) $v = \pi \det(\mathbf{B})$.
```

## REFERENCES

[1] A. Beck and Y. C. Eldar, *Strong duality in nonconvex quadratic optimization with two quadratic constraints*, SIAM J. Optim., 17 (2006), pp. 844–860.

[2] Z. Ben-Haim and Y. C. Eldar, *Maximum set estimators with bounded estimation error*, IEEE Trans. Signal Process., 53 (2005), pp. 3172–3182.

[3] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, MPS SIAM Ser. Optim. 2, SIAM, Philadelphia, 2001.

[4] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[5] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[6] S. P. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[7] Y. C. Eldar, *Minimum variance in biased estimation: Bounds and asymptotically optimal estimators*, IEEE Trans. Signal Process., 52 (2004), pp. 1915–1930.

[8] Y. C. Eldar, A. Ben-Tal, and A. Nemirovski, *Linear minimax regret estimation of deterministic parameters with bounded data uncertainties*, IEEE Trans. Signal Process., 52 (2004), pp. 2177–2188.

[9] Y. C. Eldar, A. Ben-Tal, and A. Nemirovski, *Robust mean-squared error estimation in the presence of model uncertainties*, IEEE Trans. Signal Process., 53 (2005), pp. 168–181.

[10] C. Fortin and H. Wolkowicz, *The trust region subproblem and semidefinite programming*, Optim. Methods Softw., 19 (2004), pp. 41–67.

[11] L. El Ghaoui and H. Lebret, *Robust solutions to least-squares problems with uncertain data*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1035–1064.

[12] G. H. Golub, P. C. Hansen, and D. P. O'Leary, *Tikhonov regularization and total least squares*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 185–194.

[13] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[15] P. C. Hansen, *Regularization tools, a MATLAB package for analysis of discrete regularization problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

[16] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed problems: Numerical Aspects of Linear Inversion*, SIAM Monogr. Math. Model. Comput. 4, SIAM, Philadelphia, 1997.

[17] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.

[18] T. Kailath, *Lectures on Linear Least-Squares Estimation*, Springer-Verlag, Wein, New York, 1976.

[19] M. Milanese and G. Belforte, *Estimation theory and uncertainty intervals evaluation in the presence of unknown but bounded errors: Linear families of models and estimators*, IEEE Trans. Automat. Control, 27 (1982), pp. 408–414.

[20] M. Milanese and R. Tempo, *Optimal algorithms theory for robust estimation and prediction*, IEEE Trans. Automat. Control, 30 (1985), pp. 730–738.

[21] M. Milanese and A. Vicino, *Optimal estimation theory for dynamic systems with set membership uncertainty: An overview*, Automatica J. IFAC, 27 (1991), pp. 997–1009.

[22] J. J. Moré, *Generalizations of the trust region subproblem*, Optim. Methods Softw., 2 (1993), pp. 189–209.

[23] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, Studies in Applied Mathematics 13, SIAM, Philadelphia, 1994.

[24] J. P. Norton, *Identification and application of bounded parameter models*, Automatica J. IFAC, 23 (1987), pp. 497–507.

[25] M. S. Pinsker, *Optimal filtering of square-integrable signals in Gaussian noise*, Prob. Inf. Transm., 16 (1980), pp. 120–133.

[26] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[27] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11/12 (1999), pp. 625–653.

[28] A. N. Tikhonov, *Solution of incorrectly formulated problems and the regularization method*, Soviet. Math. Dokl., 5 (1963), pp. 1035–1038.

[29] J. F. Traub, G. Wasikowski, and H. Wozinakowski, *Information-based Complexity*, Academic Press, New York, 1988.

[30] S. Xu, R. M. Freund, and J. Sun, *Solution methodologies for the smallest enclosing circle problem*, Comput. Optim. Appl., 25 (2003), pp. 283–292.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 626–642

# COVARIANCE MATRICES FOR PARAMETER ESTIMATES OF CONSTRAINED PARAMETER ESTIMATION PROBLEMS*

HANS GEORG BOCK[†], EKATERINA KOSTINA[†], AND OLGA KOSTYUKOVA[‡]

**Abstract.** In this paper we show how, based on the conjugate gradient method, to compute the covariance matrix of parameter estimates and confidence intervals for constrained parameter estimation problems as well as their derivatives.

**Key words.** PDE constrained parameter estimation and optimal experiment design problems, covariance matrix of parameter estimates, iterative linear algebra methods

**AMS subject classifications.** 65K10, 15A09, 65F30

**DOI.** 10.1137/040617893

## 1. Introduction.

Parameter estimation and optimal design of experiments are important steps in establishing models that reproduce a given process quantitatively correctly. The aim of parameter estimation is to reliably identify model parameters from sets of noisy experimental data. The "accuracy" of the parameters, i.e., their statistical distribution depending on data noise, can be estimated up to first order by means of an approximation of a covariance matrix for parameter estimates and corresponding confidence regions. In practical applications, however, one often finds that the experiments performed to obtain the required measurements are expensive, but nevertheless do not guarantee satisfactory parameter accuracy. In order to maximize the accuracy of the parameter estimates, additional experiments can be designed with optimal experimental settings or controls (e.g., initial conditions, measurement devices, sampling times, temperature profiles, feed streams, etc.) subject to constraints. As an objective functional, a suitable function of the covariance matrix (e.g., trace, determinant, maximal eigenvalue, etc.) can be used. The possible constraints in this problem describe costs, feasibility of experiments, domain of models, etc. Using Newton-type methods for solving constrained optimal design problems implies the knowledge of the derivatives of the objective function, which is, let us repeat, a function of the covariance matrix. Hence, methods of computing the covariance matrix and its derivatives are of great importance in parameter estimation and optimal design of experiments.

While a representation of the covariance matrix is well known for unconstrained parameter estimation (see, e.g., [2]) we are interested in the covariance matrix for the [illegible]. And this for a good reason! Since the seventies, general nonlinear parameter estimation problems in dynamic processes described by ordinary or partial differential-algebraic equations (DAEs) have attracted much attention from researchers. Applying the so-called boundary value problem approach [3], [4], also

---

[*] Received by the editors October 29, 2004; accepted for publication (in revised form) by L. Vandenberghe November 3, 2006; published electronically May 22, 2007. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) through SFB 359 and was partially supported by the state program of the fundamental research of Republic of Belarus "Mathematical Structures 15."

http://www.siam.org/journals/simax/29-2/61789.html

[†] IWR, University of Heidelberg, Im Neuenheimer Feld, 368, D-69120, Heidelberg, Germany (bock@iwr.uni-heidelberg.de, ekaterina.kostina@iwr.uni-heidelberg.de).

[‡] Institute of Mathematics, National Academy of Sciences of Belarus, Surganov str. 11, 220072, Minsk, Belarus (kostyukova@im.bas-net.by).

known nowadays as "all-at-once" methods, to such problems leads to [illegible] nonlinear parameter estimation problems, where the implicit constraint is treated by infeasible point methods. One of the issues of this paper is to formulate a suitable representation of the covariance matrix and confidence regions for constrained parameter estimation. Another question is how to numerically compute the covariance matrix.

So far numerical methods for parameter estimation and optimal design of experiments in dynamic processes have been based on direct linear algebra methods for computing the covariance matrix and its derivatives. The direct methods are variants of Gaussian elimination and involve an explicit matrix factorization for solving linear systems of equations. They were originally developed for systems of nonlinear ODEs or DAEs, where direct linear algebra methods are more effective for forward model problems than iterative methods. The iterative methods work by repeatedly improving an approximate solution until it is accurate enough.

On the other hand, for very large scale constrained systems with sparse matrices of special structure, e.g., originating from discretization of partial differential equations (PDEs), direct linear algebra methods based on Gauss elimination or orthogonal decompositions are not competitive with iterative linear algebra methods (see, e.g., [17]) even for forward models. Hence, in case of parameter estimation in PDE models, generalizations of iterative linear algebra methods to the computation of the covariance matrix and its derivatives are crucial for practical applications, and this defines the aim of this paper. One of the intriguing results of this paper is that by solving nonlinear constrained least squares problems by conjugate gradient methods, we get as a by-product the covariance matrix and confidence intervals as well as their derivatives.

The paper is organized as follows. In the next section we introduce constrained parameter estimation problems and sketch their solution with a generalized Gauss–Newton method. A representation of the covariance matrix and confidence regions for constrained parameter estimation problems is described in section 3. An interpretation of the covariance matrix as a solution of a linear system is given in section 4. Further, it is shown there that the columns of the covariance matrix solve special constrained quadratic problems. Based on this observation, a conjugate gradient method for constrained quadratic problems is outlined in section 5, which serves as a prototype of an iterative solution method. Section 6 describes a numerical procedure to compute the covariance matrix as a by-product of a conjugate gradient method used to solve the linearized PDE constrained parameter estimation problems at the iterations of a generalized Gauss–Newton method.

## 2. Constrained parameter estimation problems.

We consider that the model is described by a model response, a nonlinear function $M(x,t) \in \mathbb{R}$ depending on variables $x$ and time $t$. The vector $x$ includes unknown parameters and the so-called state variables, that is, the variables resulting from discretization of dynamic systems. It is assumed that, at times $t_j$, $j = 1, \ldots, m_1$, measurements $\eta_j$, $j = 1, \ldots, m_1$, are available,

$$\eta_j = M(x^{true}, t_j) + \varepsilon_j,$$

which are subject to measurement errors $\varepsilon_j$. Here $x^{true}$ denotes the "true" values of the parameters and state variables. We assume then the errors are independent and normally distributed with zero mean and variances $\sigma$. Then minimization of the weighted least squares functional

$$||F_1(x)||_2^2 := \sum_j \frac{(\eta_j - M(x,t_j))^2}{\sigma^2} \tag{2.1}$$

is known since Gauss [6] to deliver a maximum likelihood estimate. Frequently, the model is given implicitly; then the variables $x$ satisfy equality constraints

$$F_2(x) = 0.$$

For the case under consideration in this paper, $F_2$ represents a discretized PDE boundary value problem. Summing up, the nonlinear constrained parameter estimation problem can be formulated as

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & \frac{1}{2}||F_1(x)||_2^2 \\ \text{s.t.} \ & F_2(x) = 0. \end{aligned} \tag{2.2}$$

For simplicity we assume that the functions $F_i : D \subset R^n \to \mathbb{R}^{m_i}$, $i = 1, 2$, are twice-continuously differentiable. Further we assume that $m_2 < n$ and $m_1 + m_2 \geq n$. Note that the inequality $m_2 \geq n$, together with the condition $\operatorname{rank} \partial F_2(x)/\partial x = n$ $\forall\, x \in \mathcal{X} := \{x \in R^n : \ F_2(x) = 0\}$, implies that the feasible set $\mathcal{X}$ consists of isolated points.

The method of choice for the boundary value problem or all-at-once approach to solve problem (2.2) is a generalized Gauss–Newton method, because it almost shows performance of the second order method requiring only provision of the first order derivatives. According to a generalized Gauss–Newton method, a new iterate is (basically) generated by

$$x^{k+1} = x^k + t^k \Delta x^k, \ \ 0 < t^k \leq 1, \tag{2.3}$$

where the increment $\Delta x^k$ is the solution of the linearized problem at $x = x^k$:

$$\begin{aligned} \min_{\Delta x\in\mathbb{R}^n} \ & \frac{1}{2}||F_1(x) + J_1(x)\Delta x||_2^2 \\ \text{s.t.} \ & F_2(x) + J_2(x)\Delta x = 0, \end{aligned} \tag{2.4}$$

and the stepsize $t_k$ is determined by an appropriate line search. Here $J_i(x)$ denotes a Jacobian $J_i(x) = \frac{\partial F_i(x)}{\partial x}$, $i = 1, 2$.

If the Jacobians $J_1$ and $J_2$ satisfy two regularity assumptions on $D$,

$$\operatorname{rank}\ J_2(x) = m_2, \tag{2.5}$$

$$\operatorname{rank}\ J = n, \ \ J = J(x) = \begin{pmatrix} J_1(x) \\ J_2(x) \end{pmatrix}, \tag{2.6}$$

then the linearized problem (2.4) has a unique solution $\Delta x^k$ and a unique Lagrange vector $\lambda^k$ satisfying the following optimality conditions:

$$\begin{aligned} J_1^T(x)J_1(x)\Delta x^k + J_2^T(x)\lambda^k &= -J_1^T(x)F_1(x), \\ J_2(x)\Delta x^k &= -F_2(x). \end{aligned} \tag{2.7}$$

Using (2.7) one can easily show that $\Delta x^k$ can be formally written, with the help of a solution operator $J^+$, as

$$\Delta x^k = -J^+(x^k)F(x^k),\ F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \end{pmatrix}.$$

The solution operator $J^+$ is a generalized inverse, that is, it satisfies $J^+JJ^+ = J^+$ and is explicitly given by

$$J^+(x) = \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} J_1^T(x)J_1(x) & J_2(x)^T \\ J_2(x) & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1(x)^T & 0 \\ 0 & \mathbb{I} \end{pmatrix}. \tag{2.8}$$

Here $\mathbb{I}$ denotes an identity matrix. Let us note once again that the conditions (2.5) and (2.6) guarantee that the matrix

$$\begin{pmatrix} J_1^T(x)J_1(x) & J_2(x)^T \\ J_2(x) & 0 \end{pmatrix}$$

is nonsingular. By definition, $J^+$ satisfies $J^+J = \mathbb{I}$, which will be used later.

**3. Approximation of the covariance matrix and computation of confidence regions.** It is important for parameter estimation problems to compute not only parameters but also a statistical assessment of the accuracy of these parameter estimates. This can be done by means of the covariance matrix. A representation of the covariance matrix for unconstrained nonlinear parameter estimation problems is well known; see, e.g., [2]. In the following, this notion is generalized to constrained parameter estimation problems.

Let $J$ be the Jacobian at the solution $x^*$ and $J^+$ be the corresponding generalized inverse computed according to (2.8). Due to the statistical errors of the data as input of the parameter estimation problem, the estimate as the result of the solution procedure is a random variable. Indeed, the parameter estimation problem (2.2) can be rewritten in the form

$$\begin{aligned} &\min_x \ \frac{1}{2}||F_1(x,\varepsilon)||_2^2 \\ &\text{s.t.}\ \ F_2(x) = 0, \end{aligned} \tag{3.1}$$

depending on the measurement errors $\varepsilon = (\varepsilon_j, j = 1, \dots, m_1)$. Here

$$F_{1j}(x,\varepsilon) := \frac{M(x,t_j) - \eta_j}{\sigma} = \frac{M(x,t_j) - (M(x^{true},t_j) + \varepsilon_j)}{\sigma},\ j = 1,\dots,m_1,$$

with the measurement errors $\varepsilon \in \mathcal{N}(0,\sigma^2\mathbb{I})$. Consider now solution $x(\varepsilon)$ of problem (3.1) and suppose that $x(\varepsilon) \to x(0) = x^{true}$ when $\|\varepsilon\| \to 0$. For problem (3.1) the optimality conditions, together with constraints, read

$$\mathcal{F}(x,\lambda,\varepsilon) = 0,\ \mathcal{F}(x,\lambda,\ \varepsilon) := \begin{pmatrix} J_1^T(x,\varepsilon)F_1(x,\varepsilon) + J_2^T(x)\lambda \\ F_2(x) \end{pmatrix}. \tag{3.2}$$

For the error $\varepsilon = 0$ we have $x(0) = x^{true}$, $F_2(x^{true}) = 0$, and $F_1(x^{true},0) = 0$, and hence, by the regularity assumptions it follows from (3.2) that $\lambda(0) = 0$. Further, the Jacobian

$$\left.\frac{\partial \mathcal{F}(x,\lambda,\varepsilon)}{\partial(x,\lambda)}\right|_{\varepsilon=0} = \begin{pmatrix} J_1^T(x(0),0)J_1(x(0),0) & J_2(x(0))^T \\ J_2(x(0)) & 0 \end{pmatrix}$$

is nonsingular, and we may apply the implicit function theorem. According to the theorem, in the neighborhood of $\varepsilon_0 = 0$ there exist unique functions $x(\varepsilon)$, $\lambda(\varepsilon)$ satisfying (3.2) and the initial conditions $x(0) = x^{true}$ and $\lambda(0) = 0$, and the derivatives $\frac{\partial x(0)}{\partial \varepsilon} \in \mathbb{R}^{n \times m_1}$ and $\frac{\partial \lambda(0)}{\partial \varepsilon} \in \mathbb{R}^{m_2 \times m_1}$ satisfy the linear system

$$\begin{pmatrix} J_1^T(x(0),0)J_1(x(0),0) & J_2(x(0))^T \\ J_2(x(0)) & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial x(0)}{\partial \varepsilon} \\ \frac{\partial \lambda(0)}{\partial \varepsilon} \end{pmatrix} = - \begin{pmatrix} J_1^T(x(0),0)(-\frac{1}{\sigma}\mathbb{I}) \\ 0 \end{pmatrix}.$$

By Taylor expansion one obtains

$$x(\varepsilon) = x(0) + \frac{\partial x(0)}{\partial \varepsilon}\varepsilon + O(||\varepsilon||^2) = x^{true} + J^+(x^{true}) \begin{pmatrix} \frac{1}{\sigma}\varepsilon \\ 0 \end{pmatrix} + O(||\varepsilon||^2). \tag{3.3}$$

Consequently, up to the first order $x(\varepsilon)$ is normally distributed with expected value $\mathcal{E}(x(\varepsilon)) = x^{true}$ and variances

$$\mathcal{E}\Big((x(\varepsilon) - x^{true})(x(\varepsilon) - x^{true})^T\Big). \tag{3.4}$$

Thus, due to (3.3), (3.4) we may approximate a variance-covariance matrix by the following matrix which we will later call, for the sake of brevity, [illegible] :

$$\begin{aligned} C &:= \mathcal{E}\left(J^+(x) \begin{pmatrix} \frac{1}{\sigma}\varepsilon \\ 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma}\varepsilon \\ 0 \end{pmatrix}^T J^{+T}(x)\right) \\ &= J^+(x)\mathcal{E}\left(\begin{pmatrix} \frac{1}{\sigma}\varepsilon \\ 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma}\varepsilon \\ 0 \end{pmatrix}^T\right) J^{+T}(x) \\ &= J^+(x) \begin{pmatrix} \mathbb{I}_{m_1 \times m_1} & 0 \\ 0 & 0_{m_2 \times m_2} \end{pmatrix} J^{+T}(x). \end{aligned}$$

Obviously, the matrix $C$ is a symmetric positive semidefinite matrix with rank $C = \bar{m} := n - m_2$.

Now let us show how to compute confidence regions for all variables in constrained parameter estimation problems. For the unconstrained case the way of computation of confidence regions is well known, while it is less well known for the constrained case and was introduced by one of the authors in [4]. Generalizing the unconstrained case, we define a nonlinear confidence region for the solution $x^*$ of the nonlinear [illegible] parameter estimation problem (2.2) by

$$G_N(\alpha) := \{x |\, F_2(x) = 0,\ \|F_1(x)\|_2^2 - \|F_1(x^*)\|_2^2 \leq \gamma^2(\alpha)\},$$

where $\gamma^2(\alpha) := \chi^2_{\bar{m}}(1-\alpha)$ is the quantile of the $\chi^2$ distribution for value $\alpha$ with $\bar{m}$ degrees of freedom. The nonlinear confidence region $G_N(\alpha)$ can be approximated by a linearized confidence region

$$\begin{aligned} G_L(\alpha) := \{x |\ & F_2(x^*) + J_2(x^*)(x - x^*) = 0, \\ & \|F_1(x^*) + J_1(x^*)(x - x^*)\|_2^2 - \|F_1(x^*)\|_2^2 \leq \gamma^2(\alpha)\}. \end{aligned}$$

Note that since $x^*$ is optimal in (2.2), the latter expression can be rewritten as

$$G_L(\alpha) = \{x \mid J_2(x^*)(x - x^*) = 0,\ \|J_1(x^*)(x - x^*)\|_2^2 \leq \gamma^2(\alpha)\}. \tag{3.5}$$

Indeed, the optimality conditions for $x^*$ and the Lagrange vector $\lambda^*$ in problem (2.2), namely

$$F_1(x^*)^T J_1(x^*) + \lambda^{*T} J_2(x^*) = 0,$$

yield that for any $\Delta x$ satisfying $J_2 \Delta x = 0$, the following relations hold:

$$F_1(x^*)^T J_1(x^*) \Delta x = -\lambda^{*T} J_2(x^*) \Delta x = 0. \tag{3.6}$$

Taking into account relations (3.6) and the equality $F_2(x^*) = 0$ results in (3.5).

The following lemma gives another, more illustrative, representation of the linear confidence region.

LEMMA 3.1. [illegible] $x^*$ [illegible] (2.2) [illegible] $\varepsilon$ [illegible] (2.5) [illegible] (2.6) [illegible]

$$G_L(\alpha) = \bar{G}_L(\alpha) := \left\{ x^* + \Delta x \mid \Delta x = -J^+(x^*) \begin{pmatrix} \eta \\ 0 \end{pmatrix}, \ \ \|\eta\|_2^2 \le \gamma^2(\alpha) \right\}. \tag{3.7}$$

[illegible] Consider $x \in G_L(\alpha)$, and denote $\Delta x := x - x^*$. It follows from (3.5) that

$$\|J_1(x^*) \Delta x\|_2^2 \le \gamma^2(\alpha).$$

If we choose $\eta = -J_1(x^*)\Delta x$, then

$$\begin{aligned} \Delta x = J^+(x^*) J(x^*) \Delta x = J^+(x^*) \begin{pmatrix} J_1(x^*)\Delta x \\ J_2(x^*)\Delta x \end{pmatrix} = J^+(x^*) \begin{pmatrix} J_1(x^*)\Delta x \\ 0 \end{pmatrix} \\ = -J^+(x^*) \begin{pmatrix} \eta \\ 0 \end{pmatrix}. \end{aligned}$$

Hence, $x \in \bar{G}_L(\alpha)$.

Let us take $x \in \bar{G}_L(\alpha)$. Then, by definition of $J^+$ the vector $\Delta x$ satisfies the linear system

$$\begin{aligned} J_1^T(x^*) J_1(x^*) \Delta x + J_2^T(x^*) \lambda &= -J_1^T(x^*) \eta, \\ J_2(x^*) \Delta x &= 0, \end{aligned}$$

with some vector $\lambda$. Then

$$\begin{aligned} (\eta + J_1(x^*)\Delta x)^T J_1(x^*) \Delta x &= \eta^T J_1(x^*) \Delta x + \Delta x^T J_1^T(x^*) J_1(x^*) \Delta x \\ &= -\lambda^T J_2(x^*) \Delta x = 0. \end{aligned}$$

Now let us compute $||\eta + J_1(x^*)\Delta x||_2^2$ :

$$\begin{aligned} 0 \le ||\eta + J_1(x^*)\Delta x||_2^2 &= \|\eta\|_2^2 + 2\eta^T J_1(x^*)\Delta x + \Delta x^T J_1^T(x^*) J_1(x^*) \Delta x \\ &= \|\eta\|_2^2 - \|J_1(x^*)(x - x^*)\|_2^2. \end{aligned}$$

Hence, $\|J_1(x^*)\Delta x\|_2^2 \le \|\eta\|_2^2 \le \gamma^2(\alpha)$. This means that $x \in G_L(\alpha)$. ☐

The next result shows that the linearized confidence region $G_L(\alpha)$ is contained [illegible] in a minimal box, which is the cross product of so-called confidence intervals.

LEMMA 3.2. [illegible] $x^*$ [illegible] (3.1) [illegible] $\varepsilon$ [illegible] (2.5) [illegible] (2.6) [illegible]

$$G_L(\alpha) \subset \bigtimes_{i=1}^{n} [x_i^* - \theta_i, x_i^* + \theta_i],$$

[illegible] $\theta_i = C_{ii}\gamma(\alpha)$. [illegible] $C_{ii}^2$ [illegible] $C$ [illegible]

$$\max_{x \in G_L(\alpha)} |x_i - x_i^*| = \theta_i, \; i = 1, \dots, n.$$

[illegible] For each component of $x \in G_L(\alpha)$ we evaluate

$$|\Delta x_i|^2 = \left| e_i^T J^+(x^*) \begin{pmatrix} \eta \\ 0 \end{pmatrix} \right|^2 \le \left\| e_i^T J^+(x^*) \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \right\|_2^2 \|\eta\|_2^2 \le C_{ii}^2 \gamma^2(\alpha).$$

Now we want to show that this bound is [illegible] Let us now compute the maximum value of $|\Delta x_i|^2$, that is, the maximum value of the cost function in the following problem:

$$\begin{aligned} &\max_{\eta} \; \left| e_i^T J^+(x^*) \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \eta \right|^2 \\ &\text{s.t.} \; ||\eta||_2^2 \le \gamma^2(\alpha). \end{aligned}$$

Obviously, the solution in this problem is

$$\eta^{*T} = \gamma(\alpha) e_i^T J^+(x^*) \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \Big/ \left\| e_i^T J^+(x^*) \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \right\|_2,$$

and the optimal value of the cost function is equal to

$$\gamma^2(\alpha) \left\| e_i^T J^+(x^*) \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \right\|_2^2 = C_{ii}^2 \gamma^2(\alpha). \qquad \square$$

Lemma 3.2 shows that the diagonal elements of the covariance matrix play an important role in the statistical assessment of the estimates, namely they are used to compute confidence intervals.

We finish this section with an example illustrating nonlinear and linearized confidence regions and a confidence box. Consider, for simplicity of visualization, an unconstrained problem in two variables (Rosenbrock-type example):

$$\min \frac{1}{2} ||F_1(x)||_2^2, \;\; F_1(x) := \begin{pmatrix} x_1/\sigma_1 \\ (x_2 + \frac{1}{200}(x_1 - 50)^2)/\sigma_2 \end{pmatrix}, \;\; \sigma_1 = 1. \tag{3.8}$$

By choosing the parameter $\sigma_2$ we may change the nonlinearity of the problem (3.8): the smaller $\sigma_2$, the larger the nonlinearity. The solution of problem (3.8) is $x^* = (0, -12.5)^T$.

Figure 3.1 shows that linear confidence regions are quite good approximations of nonlinear confidence regions.

## 4. Covariance matrix as a solution of a linear system.

In this section we will show that the covariance matrix $C \in \mathbb{R}^{n \times n}$ for the constrained parameter estimation problem

$$C := J^+ \begin{bmatrix} \mathbb{I} & 0 \\ 0 & 0 \end{bmatrix} (J^+)^T \tag{4.1}$$

satisfies a linear system of equations with coefficients which are uniquely determined by the matrix $J$ (see (2.6)). In this section, for simplicity of notations, we omit the

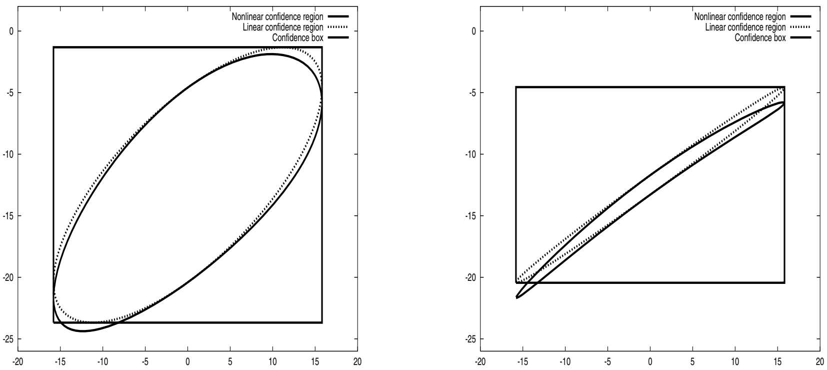


FIG. 3.1. *Nonlinear and linearized confidence regions and a confidence box for* $\gamma(\alpha) = 250$ *and for* $\sigma_2 = 0.5$ *(right) and* $\sigma_2 = 0.05$ *(left).*

dependence of $C$ on the linearization point $x$. Throughout this section we assume that the matrices $J_1$ and $J_2$ satisfy the regularity assumptions (2.5) and (2.6). Let us denote

$$M^{-1} = \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} := \begin{pmatrix} X & Y \\ Z & T \end{pmatrix},$$
$$X \in \mathbb{R}^{n\times n},\ \ Y \in \mathbb{R}^{n\times m_2},\ \ Z = Y^T \in \mathbb{R}^{m_2\times n},\ \ T \in \mathbb{R}^{m_2\times m_2}.$$

LEMMA 4.1. [illegible] $C$ (4.1)[illegible] $X$[illegible] $C \in \mathbb{R}^{n\times n}$[illegible] $Z \in \mathbb{R}^{m_2\times n}$

$$\begin{aligned} J_1^T J_1 C + J_2^T Z &= \mathbb{I}, \\ J_2 C &= 0. \end{aligned} \tag{4.2}$$

[illegible]. According to (2.8) and (4.1) we have

$$\begin{aligned} C &= \begin{pmatrix} \mathbb{I} & 0 \end{pmatrix} M^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} M^{-1} \begin{pmatrix} \mathbb{I} \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} X & Y \end{pmatrix} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X \\ Z \end{pmatrix} = X J_1^T J_1 X. \end{aligned} \tag{4.3}$$

Since the blocks $X$ and $Z$ of the matrix $M^{-1}$ satisfy the linear system

$$\begin{aligned} J_1^T J_1 X + J_2^T Z &= \mathbb{I}, \\ J_2 X &= 0, \end{aligned} \tag{4.4}$$

relation (4.3) yields

$$C = X(\mathbb{I} - J_2^T Z) = X,$$

which means that $C = X$. □

Note that, according to Lemma 4.1 the covariance matrix $C$ is a generalized inverse of the matrix $J_1^T J_1$, that is, it satisfies $C(J_1^T J_1)C = C$.

LEMMA 4.2. [illegible] $X^{(i)}$ [illegible] $X$ [illegible]

$$\min_{\kappa} f_i(\kappa) = \frac{1}{2}\kappa^T J_1^T J_1 \kappa - e_i^T \kappa \qquad (4.5)$$
$$\text{[illegible]} \quad J_2\kappa = 0,$$

[illegible] $Z^{(i)}$ [illegible] $Z$ [illegible] (4.5) [illegible] (4.5) [illegible]

$$f_i(X^{(i)}) = -\frac{1}{2}X_i^{(i)}.$$

[illegible] $e_i$ [illegible] $i$ [illegible] $X_i^{(i)}$ [illegible] $i$ [illegible] $X^{(i)}$.

[illegible] Multiplying system (4.4) with $e_i$ yields

$$\begin{aligned} J_1^T J_1 X^{(i)} + J_2^T Z^{(i)} &= e_i, \\ J_2 X^{(i)} &= 0. \end{aligned} \qquad (4.6)$$

Under assumptions (2.5) and (2.6) this system has a unique solution. The system (4.6) is nothing but the Karush–Kuhn–Tucker system for the problem (4.5), thus defining the optimal solution $X^{(i)}$ and the optimal Lagrange variables $Z^{(i)}$.

Next, we compute the value of the cost function at $X^{(i)}$:

$$\begin{aligned} f_i(X^{(i)}) &= \frac{1}{2}X^{(i)T} J_1^T J_1 X^{(i)} - e_i^T X^{(i)} \\ &= \frac{1}{2}(X^{(i)T} e_i - X^{(i)T} J_2^T Z^{(i)}) - e_i^T X^{(i)} = -\frac{1}{2}X_i^{(i)}. \qquad \square \end{aligned}$$

The following corollary of Lemma 4.2 shows how to compute the trace of the covariance matrix in terms of the optimal values of the problems (4.5). The trace of the covariance matrix can be used as the cost functional for design of optimal experiments [1].

COROLLARY 4.3.

$$\text{tr } C = \sum_{i=1}^{n} e_i^T X^{(i)} = -2\sum_{i=1}^{n} f_i(X^{(i)}). \qquad (4.7)$$

Using the representation of the covariance matrix as a solution of the linear system, we may derive the derivatives of the covariance matrix $C$ and the matrix $Z$ as the functions of the matrices $J_1$ and $J_2$,

$$\begin{aligned} C &= C(J_1, J_2) = X(J_1, J_2), \\ Z &= Z(J_1, J_2). \end{aligned}$$

These derivatives are needed in numerical methods for design of optimal nonlinear experiments [1]. Let $J_1(t) = J_1 + t\Delta J_1$ and $J_2(\mu) = J_2 + \mu\Delta J_2$, and compute the partial derivatives

$$\begin{aligned}
\frac{\partial X(J_1(t),J_2(\mu))}{\partial t}\,\Big|_{t=0,\mu=0} &= \frac{\partial X(J_1(t),J_2(\mu))}{\partial J_1}\Delta J_1\,\Big|_{t=0,\mu=0} =: L_1,\\
\frac{\partial X(J_1(t),J_2(\mu))}{\partial \mu}\,\Big|_{t=0,\mu=0} &= \frac{\partial X(J_1(t),J_2(\mu))}{\partial J_2}\Delta J_2\,\Big|_{t=0,\mu=0} =: L_2,\\
\frac{\partial Z(J_1(t),J_2(\mu))}{\partial t}\,\Big|_{t=0,\mu=0} &= \frac{\partial Z(J_1(t),J_2(\mu))}{\partial J_1}\Delta J_1\,\Big|_{t=0,\mu=0} =: R_1,\\
\frac{\partial Z(J_1(t),J_2(\mu))}{\partial \mu}\,\Big|_{t=0,\mu=0} &= \frac{\partial Z(J_1(t),J_2(\mu))}{\partial J_2}\Delta J_2\,\Big|_{t=0,\mu=0} =: R_2.
\end{aligned}$$

Differentiating the linear system (4.2) with respect to $t$ yields

$$\begin{aligned}
(\Delta J_1^T J_1 + J_1^T \Delta J_1)X + J_1^T J_1 L_1 + J_2^T R_1 &= 0,\\
J_2 L_1 &= 0.
\end{aligned}$$

Here $X = C$ and $J_1 = J_1(0)$, $J_2 = J_2(0)$. Thus, the matrices $L_1$ and $R_1$ can be found by

$$\begin{aligned}
L_1 &= -X(\Delta J_1^T J_1 + J_1^T \Delta J_1)X,\\
R_1 &= -Z(\Delta J_1^T J_1 + J_1^T \Delta J_1)X.
\end{aligned}\tag{4.8}$$

Analogously,

$$\begin{aligned}
\Delta J_2^T Z + J_1^T J_1 L_2 + J_2^T R_2 &= 0,\\
\Delta J_2 X + J_2 L_2 &= 0,
\end{aligned}$$

and hence, the matrices $L_2$ and $R_2$ can be computed by

$$\begin{aligned}
L_2 &= -X\Delta J_2^T Z - Z^T \Delta J_2 X,\\
R_2 &= -Z\Delta J_2^T Z - T\Delta J_2 X,
\end{aligned}\tag{4.9}$$

where $T = -J_2^{+} J_1^T J_1 Z^T$.

In the case of the trace of the covariance matrix, which is one of the possible criteria for the design of the experiments, the computation of the derivatives is significantly simplified.

LEMMA 4.4.

$$\frac{\partial \mathrm{tr}\ C(J_1(t),J_2)}{\partial t} = -\sum_{i=1}^{n} X^{(i)T}(\Delta J_1^T J_1 + J_1^T \Delta J_1)X^{(i)},\tag{4.10}$$

$$\frac{\partial \mathrm{tr}\ C(J_1,J_2(\mu))}{\partial \mu} = -2\sum_{i=1}^{n} X^{(i)T}\Delta J_2 Z^{(i)},\tag{4.11}$$

[illegible] $X^{(i)}$ [illegible] $Z^{(i)}$ [illegible] $i$ [illegible] $X$ [illegible] $Z$ [illegible]. The proof follows from the relations (4.8) and (4.9) and the definition of the trace of a matrix tr $C = \sum_{i=1}^{n} e_i^T C e_i$. □

Note that, computing the derivative of the covariance matrix with respect to $J_2$ requires knowledge of the matrix $Z$.

1. The relations (4.10) and (4.11) can be derived from the duality theory for convex problems [7]. According to the theory, with the Lagrangian

$$L_i(\kappa,\pi,t,\mu)=\frac{1}{2}\kappa^T J_1(t)^T J_1(t)\kappa - e_i^T\kappa + \pi^T J_2(\mu)\kappa$$

for problem (4.5), one may compute partial derivatives of the optimal value of the cost function

$$\begin{aligned}\frac{\partial}{\partial t} f_i(X^{(i)}(J_1(t),J_2(\mu)))\Big|_{t=0,\mu=0} &= \max_{\kappa\in D_i^*}\min_{\pi\in\Lambda_i^*}\frac{\partial}{\partial t}L_i(\kappa,\pi,t,\mu)\Big|_{t=0,\mu=0},\\ \frac{\partial}{\partial \mu} f_i(X^{(i)}(J_1(t),J_2(\mu)))\Big|_{t=0,\mu=0} &= \max_{\kappa\in D_i^*}\min_{\pi\in\Lambda_i^*}\frac{\partial}{\partial \mu}L_i(\kappa,\pi,t,\mu)\Big|_{t=0,\mu=0},\end{aligned}\tag{4.12}$$

where $D_i^*\subset\mathbb{R}^n$, $\Lambda_i^*\subset\mathbb{R}^{m_2}$ denote the sets of optimal primal and dual solutions of problem (4.5) at $t=\mu=0$. From (4.7) we have that

$$\begin{aligned}\frac{\partial \mathrm{tr}C(J_1(t),J_2(\mu))}{\partial t} &= -2\sum_{i=1}^{n}\frac{\partial}{\partial t}f_i(X^{(i)}(J_1(t),J_2(\mu))),\\ \frac{\partial \mathrm{tr}C(J_1(t),J_2(\mu))}{\partial \mu} &= -2\sum_{i=1}^{n}\frac{\partial}{\partial \mu}f_i(X^{(i)}(J_1(t),J_2(\mu))),\end{aligned}$$

where $X^{(i)}(J_1(t),\,J_2(\mu))$ is an optimal solution in the problem (4.5) defined at $J_1(t)$ and $J_2(\mu)$, and $f_i(X^{(i)}(J_1(t),\,J_2(\mu)))$ denotes the optimal value of the cost function. In our case, by assumptions (2.5), (2.6) each set consists of only one element, namely

$$D_i^*=\{X^{(i)}\},\Lambda_i^*=\{-Z^{(i)}\}.\tag{4.13}$$

Hence, (4.12) results in (4.10) and (4.11). Let us note that relations (4.10) and (4.11) are true only under regularity conditions (2.5) and (2.6), while the more general conditions (4.12) hold also in case of violation of (2.5) and (2.6).

2. Let us note that $C=X$ is just an identity and does not suggest a good computation. The way to compute the matrix $C$ will be described later.

**5. Conjugate gradient method for constrained quadratic problems.** We saw that the columns $X^{(i)}$, $i=1,\dots,n$, of the covariance matrix (4.1), as well as the solution of problem (2.4), solve quadratic problems that can be formally written as

$$\begin{aligned}\min_{y\in\mathbb{R}^n} f(y) &= \frac{1}{2}y^T J_1^T J_1 y + b^T y\\ \text{s.t.}\quad J_2 y &= a\end{aligned}\tag{5.1}$$

for some vectors $a$ and $b$. As in previous sections, we assume that the matrices $J_2\in\mathbb{R}^{m_2\times n}$, $J_1\in\mathbb{R}^{m_1\times n}$ satisfy regularity conditions (2.5) and (2.6).

Let us apply the classical conjugate gradient method for solving the constrained problem (5.1). The traditional way is to project the conjugate gradient method onto the null-space of $J_2$; see, e.g., [5], [9], [13], [14]. Let $\mathcal{P}\in\mathbb{R}^{n\times n}$ denote a projector onto the null-space of the matrix $J_2$,

$$\mathcal{P}=\mathbb{I}-J_2^T J_2^+,\ J_2^+=(J_2J_2^T)^{-1}J_2,\tag{5.2}$$

where $J_2^+$ is a Moore–Penrose pseudoinverse of $J_2^T$. Let us note that the matrix $J_2J_2^T$ is nonsingular because of (2.5). The algorithm can be formulated as follows.

Algorithm 1.

**Input.** [illegible] $y_0$, [illegible] $J_2 y_0 = a$ [illegible] $k = 0$, [illegible] $\nabla f(y_0) = J_1^T J_1 y_0 + b$ [illegible]

$$p_1 = -\mathcal{P}\nabla f(y_0) \in \mathbb{R}^n.$$

**Step 1.** [illegible] $p_{k+1} = 0$ [illegible] $y_k$ [illegible] (5.1) [illegible] 2

**Step 2.** [illegible]

$$\alpha_{k+1} = -\frac{\nabla f^T(y_k) p_{k+1}}{||J_1 p_{k+1}||^2}, \quad y_{k+1} = y_k + \alpha_{k+1} p_{k+1} \tag{5.3}$$

[illegible]

$$p_{k+2} = -\mathcal{P}\nabla f(y_{k+1}) + \frac{||\mathcal{P}\nabla f(y_{k+1})||^2}{||\mathcal{P}\nabla f(y_k)||^2} p_{k+1}.$$

[illegible] $k = k + 1$ [illegible] 1

In exact arithmetic the method converges in at most $\bar{m} = n - m_2$ steps to the solution of the problem (5.1).

During the iterations (5.3) we need to compute projections $\mathcal{P}\nabla f(y_k)$ for each $k = 0, 1, 2, \ldots,$

$$\mathcal{P}\nabla f(y_k) = \nabla f(y_k) - J_2^T J_2^+ \nabla f(y_k) = \nabla f(y_k) - J_2^T u_k,$$

where

$$u_k := J_2^+ \nabla f(y_k). \tag{5.4}$$

We will use the fact that the vector $u_k$ is a solution of an unconstrained least squares problem

$$\min_{u \in \mathbb{R}^{m_2}} g(u) = \frac{1}{2} ||J_2^T u - r||^2 \tag{5.5}$$

with $r = \nabla f(y_k)$; see, e.g., [14].

Consequently, in order to compute the vector $u_k$ (5.4) we need to solve problem (5.5). This can be done again with the conjugate gradient method for the unconstrained least squares problem, which can be summarized as follows.

Algorithm 2.

**Input.** [illegible] $v_0$, [illegible] $l = 0$, [illegible] $q_1 = -\nabla g(v_0) = -J_2(J_2^T v_0 - r)$.

**Step 1.** [illegible] $q_{l+1} = 0$ [illegible] $v_l$ [illegible] (5.5) [illegible] 2

**Step 2.** [illegible]

$$\alpha_{l+1} = -\frac{\nabla g^T(v_l) q_{l+1}}{||J_2^T q_{l+1}||^2}, \quad v_{l+1} = v_l + \alpha_{l+1} q_{l+1}, \tag{5.6}$$

$$q_{l+2} = -\nabla g(v_{l+1}) + \frac{||\nabla g(v_{l+1})||^2}{||\nabla g(v_l)||^2} q_{l+1}.$$

[illegible] $l = l + 1$ [illegible] 1

Let us note that the conjugate gradient Algorithm 1 for solving the constrained quadratic problem (5.1), which we present here following [14], is equivalent to the preconditioned conjugate gradient method with residual update discussed in detail in [9].

We do not want to solve the problem (5.5) for each $\nabla f(y_k)$, $k = 2, 3, \dots, \bar{m}$ in order to construct vectors $J_2^T u_k$, but rather to use the results of some previous calculations. How are we to do this? Suppose we have solved the problem (5.5) for some $r = \nabla f(y_k)$ in $m_2$ iterations. As a result we have $m_2$ vectors

$$q_1, q_2, \dots, q_{m_2} \in \mathbb{R}^{m_2}, \tag{5.7}$$

which are linearly independent and are $J_2 J_2^T$ conjugate:

$$\begin{aligned} &q_i^T J_2 J_2^T q_j = 0,\ i \neq j, \\ &q_i^T J_2 J_2^T q_i \neq 0,\ i = 1, \dots, m_2. \end{aligned}$$

In what follows we show that

$$J_2^+ = \Big(q_1, \dots, q_{m_2}\Big) \operatorname{diag}(\beta_i, i = 1, \dots, m_2) \Big(q_1, \dots, q_{m_2}\Big)^T J_2, \tag{5.8}$$

where the numbers are computed by $\beta_i = 1/||J_2^T q_i||^2 \neq 0, i = 1, \dots, m_2$. Indeed, by construction,

$$\Big(q_1, \dots, q_{m_2}\Big)^T J_2 J_2^T \Big(q_1, \dots, q_{m_2}\Big) = \operatorname{diag}(\beta_i^{-1}, i = 1, \dots, m_2).$$

Then, obviously,

$$\begin{aligned} (5.9)\ &\Big(q_1, \dots, q_{m_2}\Big) \operatorname{diag}(\beta_i, i = 1, \dots, m_2) \Big(q_1, \dots, q_{m_2}\Big)^T \\ &= \Big(q_1, \dots, q_{m_2}\Big) \Big[\Big(q_1, \dots, q_{m_2}\Big)^T J_2 J_2^T \Big(q_1, \dots, q_{m_2}\Big)\Big]^{-1} \Big(q_1, \dots, q_{m_2}\Big)^T = (J_2 J_2^T)^{-1}, \end{aligned}$$

since the matrices $(q_1, \dots, q_{m_2})$ and $J_2 J_2^T$ are nonsingular. The formula (5.8) follows immediately from (5.9) and the representation of $J_2^+$. Thus, there is no necessity to solve the problem (5.5) for each $\nabla f(y_k)$, $k = 2, 3, \dots, \bar{m}$, in order to construct the vectors $J_2^T u_k$. The vectors $J_2^T u_k$ may be computed as

$$\begin{aligned} &J_2^T u_k = J_2^T J_2^+ \nabla f(y_k) \\ &= \Big(J_2^T q_1, \dots, J_2^T q_{m_2}\Big) \operatorname{diag}(1/||J_2^T q_i||^2,\ i = 1, \dots, m_2) \Big(J_2^T q_1, \dots, J_2^T q_{m_2}\Big)^T \nabla f(y_k), \end{aligned}$$

using the vectors $J_2^T q_i$ and the numbers $||J_2^T q_i||^2$, $i = 1, \dots, m_2$, that are constructed during the solution process of only one problem (5.5). This means that to compute each vector $J_2^T u_k$ we need two matrix-vector multiplications.

[illegible] 3. Assume that the solution process for solving problem (5.5) terminates after $k < m_2$ iterations, that is, we do not have the complete set of the vectors (5.7). This means that at the next iteration of Algorithm 1 we have to solve problem (5.5) again with the new right-hand side $r = r^{new}$. We may use already computed conjugate vectors $q_1, q_2, \dots, q_k$ from the previous iteration of Algorithm 1 and may start the solution process (5.6) for $l = k + 1, \dots$ with the vector

$$v_k = (q_1, q_2, \dots, q_k) \operatorname{diag}(\beta_1, \dots, \beta_k) (q_1, q_2, \dots, q_k)^T J_2 r^{new}.$$

As a result we get the sequence of the conjugate vectors

$$q_1, q_2, \dots, q_k, q_{k+1}, \dots, q_{\bar{k}}, \ \bar{k} \geq k. \tag{5.10}$$

At the next iteration of Algorithm 1 we proceed with the same procedure.

Let us summarize the properties of the constructed vectors $y_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{m_2}$, $p_k \in \mathbb{R}^n$, $k = 1, \dots, \bar{m}$.

- The vectors $y_k$ and $-u_k$ are optimal primal and dual solutions in the problem

$$\min f(y), y \in y_0 + \mathrm{Span}\{p_1, \dots, p_k\}.$$

- $\mathcal{P}p_k = p_k$ and, hence, $J_2 p_k = 0$, meaning that the generated iterates remain in the null-space of the constraints.
- The vectors $p_1, \dots, p_{\bar{m}}$ are linearly independent.
- The vectors $p_1, \dots, p_{\bar{m}}$ form a basis of the null-space of the matrix $J_2$.
- The following relations are true:

$$p_i^T \mathcal{P} J_1^T J_1 \mathcal{P} p_j = p_i^T J_1^T J_1 p_j = \begin{cases} = 0, & i \neq j \\ \neq 0, & i = j. \end{cases} \tag{5.11}$$

[illegible] 4. Algorithms 1 and 2 are known to terminate in at most $\bar{m}$ and $m_2$ iterations in exact arithmetic. Roundoff errors destroy the conjugacy among search directions and finite termination after $\bar{m}$ and $m_2$ iterations usually will not appear. Detailed discussion of convergence aspects and influence of roundoff errors for conjugate gradient methods can be found in [5], [10], [12], [16], and many other references; see also surveys [8], [11], [15].

[illegible] 5. The other possibilities of computing projections without using iterative linear algebra methods are discussed in [9]. However, comparative analysis in [9] shows that the method discussed here (which is equivalent to a conjugate gradient method with residual update from [9]) is one of the most stable and robust against roundoff errors.

**6. Computation of the covariance matrix with the conjugate gradient method.** In this section we show how to use the information received in the course of the conjugate gradient method applied to (5.1) for computation of the covariance matrix $C$ (4.1).

Lemma 6.1. [illegible] $J_1$ [illegible] $J_2$ [illegible] (2.5) [illegible] (2.6) [illegible] (5.1) [illegible] $a$ [illegible] $b$ [illegible] (2.4) [illegible] $\Delta x^k$ [illegible] $p_1, \dots, p_{\bar{m}}$ [illegible] $||J_1 p_1||^2, \dots, ||J_1 p_{\bar{m}}||^2$ [illegible] (4.1) [illegible]

$$X = C = Q \mathrm{diag}(\gamma_i, i = 1, \dots, \bar{m}) Q^T, \tag{6.1}$$

[illegible]

$$Q = (p_1, \dots, p_{\bar{m}}) \in \mathbb{R}^{n \times \bar{m}}, \ \gamma_i = 1/||J_1 p_i||^2, \ i = 1, \dots, \bar{m}. \tag{6.2}$$

[illegible] Under assumptions (2.5) and (2.6) the system (4.2) has a unique solution. We want to show that the matrix $X$ (6.1) and the matrix $Z$,

$$Z = J_2^{+}(-J_1^T J_1 C + \mathbb{I}), \tag{6.3}$$

satisfy the linear system (4.2). Here again $J_2^+$ is a Moore–Penrose pseudoinverse of $J_2^T$.

Indeed,

$$J_2 X = J_2 Q \text{diag}(\gamma_i, i = 1, \dots, \bar{m}) Q^T = 0,$$

since $J_2 p_k = 0$ for all $k = 1, \dots, \bar{m}$, or in other words $J_2 Q = 0$. Now we want to show that the matrices $X$ (6.1) and $Z$ (6.3) satisfy the first equation in (4.2), namely that $J_1^T J_1 X + J_2^T Z = \mathbb{I}$. For this purpose we show that the matrix $N = 0$, where the matrix $N$ is defined by

$$N := J_1^T J_1 X + J_2^T Z - \mathbb{I} = J_1^T J_1 X + J_2^T J_2^+ (-J_1^T J_1 X + \mathbb{I}) - \mathbb{I}. \tag{6.4}$$

We represent the matrix $N$ in the form

$$N = (\mathbb{I} - J_2^T J_2^+) N + J_2^T J_2^+ N \tag{6.5}$$

and verify that

$$J_2^T J_2^+ N = 0, \tag{6.6}$$

$$(\mathbb{I} - J_2^T J_2^+) N = \mathcal{P} N = 0. \tag{6.7}$$

Indeed,

$$\begin{aligned} J_2^T J_2^+ N &= J_2^T J_2^+ J_1^T J_1 X + J_2^T J_2^+ J_2^T J_2^+ (-J_1^T J_1 X + \mathbb{I}) - J_2^T J_2^+ \\ &= J_2^T J_2^+ J_1^T J_1 X + J_2^T J_2^+ (-J_1^T J_1 X + \mathbb{I}) - J_2^T J_2^+ = 0, \end{aligned}$$

since $J_2^+ J_2^T = \mathbb{I}$. Next we show that (6.7) holds. Since the matrix $\mathcal{P}$ is a projector to the null-space of the matrix $J_2$, and the vectors $p_k$, $k = 1, \dots, \bar{m}$, form the basis of the null-space of the matrix $J_2$, then $\mathcal{P} N = 0$ if and only if $p_k^T N = 0$ for all $k = 1, \dots, \bar{m}$, or in other words $Q^T N = 0$. Hence, we need to validate $Q^T N = 0$:

$$\begin{aligned} Q^T N &= Q^T J_1^T J_1 Q \text{diag}(\gamma_i, i = 1, \dots, \bar{m}) Q^T - Q^T \\ &= \text{diag}\left(\frac{1}{\gamma_i}, i = 1, \dots, \bar{m}\right) \text{diag}(\gamma_i, i = 1, \dots, \bar{m}) Q^T - Q^T = 0. \qquad \square \end{aligned}$$

The lemma means that the covariance matrix $C \in \mathbb{R}^{n \times n}$ is uniquely defined according to formula (6.1) by the matrix $Q \in \mathbb{R}^{n \times \bar{m}}$ and the numbers $||J_1 p_i||^2$, $i = \overline{1, \bar{m}}$. Thus, we can store the matrix $Q$ and the numbers (6.2) instead of the matrix $C$.

[illegible] 6. The proof of Lemma 6.1 implies that to compute the covariance matrix $C$ by the formula (6.1) we may use any set of vectors $\bar{p}_1, \dots, \bar{p}_{\bar{m}}$, which satisfy the following conditions:

a) $\bar{p}_1, \dots, \bar{p}_{\bar{m}}$, are linearly independent;
b) $J_2 \bar{p}_k = 0,\ k = 1, \dots, \bar{m}$;
c) $\bar{p}_i^T J_1^T J_1 \bar{p}_j = 0,\ i \neq j;\ \bar{p}_i^T J_1^T J_1 \bar{p}_i,\ \neq 0,\ i, j = 1, \dots, \bar{m}$.

Let us now show how to compute the matrix $Z$. According to (6.3), in order to compute $Z$ we need to know $J_2^+$. In the case of the conjugate gradient method being applied to problem (5.1) we have, in a general case, the complete sequence of the vectors (5.7) as a result of solving problem (5.5). Using these vectors we can compute the matrix $J_2^+$ according to (5.8).

In order to compute the partial derivatives of the functions of the covariance matrix that are used in the design of optimal experiments [1] one can use the representation (6.1) of the covariance matrix and the formulas (4.8) and (4.9). For example, the derivatives of the trace of the covariance matrix have the following representation as a consequence of Lemma 4.4.

COROLLARY 6.2. $\frac{\partial \text{tr } C(J_1(t),J_2)}{\partial t}$ $\frac{\partial \text{tr } C(J_1,J_2(\mu))}{\partial \mu}$ (4.10) (4.11) $X^{(i)} = Q\text{diag}(\gamma_i, i = 1,\dots,\bar{m})Q_{i\cdot}$, $Q_{i\cdot}$ $i$ $Q \in \mathbb{R}^{\bar{m}\times n}$ $Z^{(i)} = J_2^+(-J_1^T J_1 X^{(i)} + e_i)$, $i = 1,\dots,n$.

7. As we noted before in the case of roundoff errors in Algorithms 1 and 2, the first $\bar{m}$ vectors $p_1,\dots,p_{\bar{m}}$ may be computed with numerical errors, and this leads to numerical errors in (6.1)–(6.3) and (4.8) for computing the covariance matrix $C$ and its derivatives. Analysis and estimation of such errors, as well as the development of approximate methods for computation of the matrix $C$, is a very important topic, but not the issue of this paper. This issue can be addressed using results from [5], [10], [12], [16], the formulas (6.1)–(6.3), and Remark 6.

Let us note further that according to Lemma 4.2 the columns $X^{(i)}$, $i = 1,\dots,n$, of the covariance matrix $C$ may be computed by applying conjugate gradient methods directly to problem (4.5). In this case all $n$ problems may be computed in parallel independently from each other until desired accuracy of the resulting vector $X^{(i)}$.

## 7. Conclusions.

For solving constraint parameter estimation and optimal design problems, we need the knowledge of the covariance matrix of the parameter estimates and its derivatives. Hence, development of effective methods for presentation and computation of the covariance matrix and its derivatives, based on iterative methods, is crucial for practical applications, which is the aim of this paper. In this paper, we have given suitable representations for the covariance matrix and confidence intervals as well as its derivatives and have shown that by solving nonlinear constrained least squares problems by conjugate gradient methods we get, as a by-product, these matrices and confidence intervals practically for free. The results can be generalized to other Krylov-type methods. This paper is the first of a series and its results are of a more theoretical nature. The forthcoming research will be devoted to numerical aspects including choice of effective preconditioners and effective implementation of the described methods for parameter estimation and design of optimal parameters in processes defined by PDEs.

**Acknowledgments.** We thank the anonymous referees and Lieven Vandenberghe whose comments and suggestions helped to improve the presentation of the results.

## REFERENCES

[1] I. BAUER, H. G. BOCK, S. KÖRKEL, AND J. P. SCHLÖDER, *Numerical methods for optimum experimental design in DAE systems*, J. Comput. Appl. Math., 120 (2000), pp. 1–25.

[2] J. V. BECK AND K. J. ARNOLD, *Parameter Estimation in Engineering and Science,* Wiley, New York, 1977.

[3] H. G. BOCK, *Numerical treatment of inverse problems in chemical reaction kinetics*, in Modelling of Chemical Reaction Systems, K. H. Ebert, P. Deuflhard, and W. Jäger, eds., Springer Ser. Chem. Phys. 18, Springer, Heidelberg, 1981, pp. 102–125.

[4] H. G. BOCK, *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen*, Bonner Math. Schriften 183, Bonn, 1987.

[5] T. F. COLEMAN AND A. VERMA, *A preconditioned conjugate gradient approach to linear equality constrained minimization*, Comput. Optim. Appl., 20 (2001), pp. 61–72.

[6] C. F. Gauss, *Theory of the Combination of Observations Least Subject to Errors,* Original with Translation, Classics in Appl. Math. 11, SIAM, Philadelphia, 1995.

[7] E. G. Golstein, *Theory of Convex Programming*, AMS, Providence, RI, 1972.

[8] G. H. Golub and D. P. O'Leary, *Some history of the conjugate gradient and Lanczos algorithms:* 1948–1976*,* SIAM Rev., 31 (1989), pp. 50–102.

[9] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.

[10] A. Greenbaum, *Iterative Methods for Solving Linear Systems,* SIAM, Philadelphia, 1997.

[11] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995.

[12] J. Notay, *On convergence rate of the conjugate gradients in presence of rounding errors,* Numer. Math., 65 (1993), pp. 301–317.

[13] B. T. Polyak, *The conjugate gradient method in extremal problems,* USSR Comput. Math. Math. Phys., 9 (1969), pp. 94–112.

[14] B. N. Pshenichny and Yu. M. Danilin, *Numerical Methods in Extremal Problems,* MIR Publishers, Moscow, 1978.

[15] J. Saad and H. A. van der Vorst, *Iterative solution of linear systems in the* 20*th century,* J. Comput. Appl. Math., 123 (2000), pp. 1–33.

[16] A. van der Sluis and H. A. van der Vorst, *The rate of convergence of conjugate gradients,* Numer. Math., 48 (1986), pp. 543–560.

[17] H. A. van der Vorst, *Parallel iterative solution methods for linear systems arising from discretized PDE's,* in Special Course on Parallel Computing in CFD. AGARD-R-807, AGARD, Neuilly-sur-Seine, France, Workshop Lecture Notes, 1995.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 643–655

# COMBINED PERTURBATION BOUNDS: I. EIGENSYSTEMS AND SINGULAR VALUE DECOMPOSITIONS*

WEN LI† AND WEIWEI SUN‡

**Abstract.** In this paper we present some new combined perturbation bounds of eigenvalues and eigensubspaces for a Hermitian matrix $H$, particularly in an asymptotic sense, $\delta_{12}^2\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2+\sum_{i=1}^r(\lambda_i-\widetilde{\lambda}_i)^2\le\|\Delta HU_1\|_F^2+O(\|\Delta HU_1\|_F^4)$, where $\lambda_i$ denotes the eigenvalues of $H$ and $U_1$ the eigensubspace corresponding to the eigenvalues $\lambda_i$, $i=1,2,\dots,r$. The bound for each factor of eigensystems is optimal due to the sin$\Theta$ theorem and the Hoffman–Wielandt theorem. In addition, combined perturbation bounds for singular value decompositions and combined perturbation bounds in some, more general, measures are also obtained.

**Key words.** eigensystems, singular subspace, singular value, combined perturbation bound

**AMS subject classifications.** 65F10, 15A45

**DOI.** 10.1137/060648969

## 1. Introduction.

Let $C^{m\times n}$ denote the set of complex $m\times n$ matrices, $A^*$ stand for the conjugate transpose of a matrix $A$, $\lambda(A)$ be the spectrum of $A$, and $\Re(A)$ be the column space of $A$. The Frobenius norm and spectral norm of a matrix $A$ are denoted by $\|A\|_F$ and $\|A\|_2$, respectively.

Let $H$ and $\widetilde{H}$ be two $n\times n$ Hermitian matrices with the following eigendecompositions:

$$\begin{aligned} H&=\begin{pmatrix} U_1 & U_2\end{pmatrix}\begin{pmatrix}\Lambda_1 & 0\\ 0 & \Lambda_2\end{pmatrix}\begin{pmatrix}U_1^*\\ U_2^*\end{pmatrix},\\ \widetilde{H}&=\begin{pmatrix} \widetilde{U}_1 & \widetilde{U}_2\end{pmatrix}\begin{pmatrix}\widetilde{\Lambda}_1 & 0\\ 0 & \widetilde{\Lambda}_2\end{pmatrix}\begin{pmatrix}\widetilde{U}_1^*\\ \widetilde{U}_2^*\end{pmatrix},\end{aligned}\tag{1.1}$$

where $U=\begin{pmatrix}U_1 & U_2\end{pmatrix}$, $\widetilde{U}=\begin{pmatrix}\widetilde{U}_1 & \widetilde{U}_2\end{pmatrix}$ are unitary, and

$$\begin{aligned}\Lambda_1&=\operatorname{diag}(\lambda_1,\lambda_2,\dots,\lambda_r), & \Lambda_2&=\operatorname{diag}(\lambda_{r+1},\lambda_{r+2},\dots,\lambda_n),\\ \widetilde{\Lambda}_1&=\operatorname{diag}(\widetilde{\lambda}_1,\widetilde{\lambda}_2,\dots,\widetilde{\lambda}_r), & \widetilde{\Lambda}_2&=\operatorname{diag}(\widetilde{\lambda}_{r+1},\widetilde{\lambda}_{r+2},\dots,\widetilde{\lambda}_n).\end{aligned}\tag{1.2}$$

Let

$$\delta_{ij}^{(k,l)}=\min_{\lambda\in\lambda(\Lambda_i),\widetilde{\lambda}\in\lambda(\widetilde{\Lambda}_j)}\frac{|\lambda-\widetilde{\lambda}|}{|\lambda|^k|\widetilde{\lambda}|^l},\quad i,j=1,2,\tag{1.3}$$

where $k$ and $l$ are nonnegative real numbers. For simplicity, we always use the notation $\delta_{ij}=\delta_{ij}^{(0,0)}$.

---

[*] Received by the editors January 4, 2006; accepted for publication (in revised form) by R. Mathias November 17, 2006; published electronically May 22, 2007.
http://www.siam.org/journals/simax/29-2/64896.html

[†] School of Mathematical Sciences, South China Normal University, Guangzhou, 510631, People's Republic of China (liwen@scnu.edu.cn). The work of this author was supported in part by National Natural Science Foundation of China (grant 10671077) and Guangdong Provincial Natural Science Foundation (grants 031496 and 06025061), People's Republic of China.

[‡] Department of Mathematics, City University of Hong Kong, Kowloon, Hong Kong, People's Republic of China (maweiw@math.cityu.edu.hk). The work of this author was supported in part by the CityU research grant 7001770.

The perturbation bounds for eigensystems, eigenspaces, and eigenvalues have been studied by many authors; e.g., see [1, 2, 3, 4, 6, 7, 8, 9, 10, 12]. The perturbation of the eigenspace $\Re(U_1)$ is measured by the canonical angle between the subspaces $\Re(U_1)$ and $\Re(\widetilde{U}_1)$ (e.g., see [8]), defined by

$$\Theta(U_1, \widetilde{U}_1) = arc\cos(U_1^* \widetilde{U}_1 \widetilde{U}_1^* U_1)^{1/2}.$$

The classical perturbation bound for the subspace was given as the $Sin\Theta$ theorem by Davis and Kahan [3].

Theorem A (**sin**$\Theta$ theorem [3]). [illegible] $H$ [illegible] $\widetilde{H} = H + \Delta H$ [illegible] (1.1)–(1.2) [illegible]

$$\delta_{12} \| \sin \Theta(U_1, \widetilde{U}_1) \|_F \le \|R\|_F, \tag{1.4}$$

[illegible] $R = \widetilde{H} U_1 - U_1 \Lambda_1 = \Delta H U_1$

The corresponding perturbation bound for eigenvalues is

$$\sum_{i=1}^{r} (\lambda_i - \widetilde{\lambda}_i)^2 \le \|R\|_F^2 . \tag{1.5}$$

When $r = n$, the bound (1.5) is the well-known Hoffman–Wielandt theorem [6].

The perturbation bounds in (1.4) and (1.5) are given in the absolute measure $\|R\|_F$. Recently, a relative-type perturbation bound was introduced. A general form of the relative bound is

$$\| \sin \Theta(U_1, \widetilde{U}_1) \|_F \le \alpha_{lk} \| H^{-l} \Delta H \widetilde{H}^{-k} \|_F , \tag{1.6}$$

where $\alpha_{lk}$ is a positive real number. Dopico, Moro, and Molera [4], Chen and Li [2], Li [8], and Londre and Rhee [9] studied the bound for $l = k = 1/2$, and Ipsen [7] studied it for the more general case.

In this paper we focus on perturbation bounds in a combined form of eigenspaces and eigenvalues. In particular, we shall show the new perturbation bound

$$\delta_{12}^2 \| \sin \Theta(U_1, \widetilde{U}_1) \|_F^2 + (1 - \| \sin \Theta(U_1, \widetilde{U}_1) \|_2^2) \sum_{i=1}^{r} (\lambda_i - \widetilde{\lambda}_i)^2 \le \|R\|_F^2 , \tag{1.7}$$

which, in an asymptotic sense, leads to

$$\delta_{12}^2 \| \sin \Theta(U_1, \widetilde{U}_1) \|_F^2 + \sum_{i=1}^{r} (\lambda_i - \widetilde{\lambda}_i)^2 \le \|R\|_F^2 + O(\|R\|_F^4), \tag{1.8}$$

where $\delta_{12} > 0$. The bounds (1.7) and (1.8) contain both the bound for eigenspaces and the bound for eigenvalues. In comparison with Davis and Kahan's theorem, (1.7) is sharper than the bound in Davis and Kahan's theorem and it also leads to the Hoffman–Wielandt theorem. On the other hand, the bound in (1.4) can be calculated when $\lambda_i$, $\widetilde{\lambda}_i$, and $\|R\|_F$ are known. In this case, a more precise bound for eigenspaces can be obtained from (1.7). In addition, we have obtained some new bounds in a relative sense and extensions to perturbation bounds for singular values and singular subspaces.

**2. Combined bounds for eigensystems.** In this section we study combined perturbation bounds for eigensystems.

LEMMA 2.1 (see [5]). [illegible] $T \in C^{n\times n}$ [illegible] $\Lambda_i = \mathrm{diag}(\lambda_1^{(i)}, \ldots, \lambda_n^{(i)}) \in C^{n\times n}$, $i = 1, 2, 3, 4$ [illegible] $\tau$ [illegible] $\langle n \rangle$ [illegible]

$$\sigma_n^2(T) \sum |\lambda_i^{(1)}\lambda_{\tau(i)}^{(2)} - \lambda_i^{(3)}\lambda_{\tau(i)}^{(4)}|^2 \le ||\Lambda_1 T \Lambda_2 - \Lambda_3 T \Lambda_4||_F^2, \tag{2.1}$$

[illegible] $\sigma_n(T)$ [illegible] $T$

We have our main theorem below.

THEOREM 2.2. [illegible] $H$ [illegible] $\widetilde{H} = H + \Delta H$ [illegible] $n \times n$ [illegible] (1.1)–(1.2) [illegible]

$$(\delta_{12}^2 - \delta_{11}^2)\|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 + r\delta_{11}^2 \le ||R||_F^2 \tag{2.2}$$

[illegible]

$$\delta_{12}^2\|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 + (1 - \|\sin\Theta(U_1, \widetilde{U}_1)\|_2^2)\sum_{i=1}^{r}(\lambda_i - \widetilde{\lambda}_i)^2 \le ||R||_F^2. \tag{2.3}$$

[illegible] Left- and right-multiplying the equation $\widetilde{H} - H = \Delta H$ by $\widetilde{U}^*$ and $U_1$, respectively, leads to

$$\widetilde{\Lambda}\widetilde{U}^*U_1 - \widetilde{U}^*U_1\Lambda_1 = \widetilde{U}^*\Delta H U_1,$$

and in the block form,

$$\begin{pmatrix} \widetilde{\Lambda}_1\widetilde{U}_1^*U_1 - \widetilde{U}_1^*U_1\Lambda_1 \\ \widetilde{\Lambda}_2\widetilde{U}_2^*U_1 - \widetilde{U}_2^*U_1\Lambda_1 \end{pmatrix} = \widetilde{U}^*\Delta H U_1.$$

It follows that

$$\|\widetilde{\Lambda}_2\widetilde{U}_2^*U_1 - \widetilde{U}_2^*U_1\Lambda_1\|_F^2 + \|\widetilde{\Lambda}_1\widetilde{U}_1^*U_1 - \widetilde{U}_1^*U_1\Lambda_1\|_F^2 = ||\Delta H U_1||_F^2\,. \tag{2.4}$$

Since

$$\left|\left(\widetilde{\Lambda}_2\widetilde{U}_2^*U_1 - \widetilde{U}_2^*U_1\Lambda_1\right)_{ij}\right|^2 = (\widetilde{\lambda}_{i+r} - \lambda_j)^2|(\widetilde{U}_2^*U_1)_{ij}|^2 \ge \delta_{12}^2|(\widetilde{U}_2^*U_1)_{ij}|^2$$
$$i = 1, 2, \ldots, n-r;\ j = 1, 2, \ldots, r,$$

and

$$\left|\left(\widetilde{\Lambda}_1\widetilde{U}_1^*U_1 - \widetilde{U}_1^*U_1\Lambda_1\right)_{ij}\right|^2 = (\widetilde{\lambda}_i - \lambda_j)^2|(\widetilde{U}_1^*U_1)_{ij}|^2 \ge \delta_{11}^2|(\widetilde{U}_1^*U_1)_{ij}|^2,$$
$$i, j = 1, 2, \ldots, r,$$

we have

$$\delta_{12}^2\|\widetilde{U}_1^*U_2\|_F^2 + \delta_{11}^2\|\widetilde{U}_1^*U_1\|_F^2 \le \|\widetilde{\Lambda}_2\widetilde{U}_2^*U_1 - \widetilde{U}_2^*U_1\Lambda_1\|_F^2 + \|\widetilde{\Lambda}_1\widetilde{U}_1^*U_1 - \widetilde{U}_1^*U_1\Lambda_1\|_F^2\,. \tag{2.5}$$

Equation (2.2) is obtained by (2.4) and (2.5) and by noting the fact that

$$\|\widetilde{U}_1^*U_1\|_F^2 = r - \|\widetilde{U}_1^*U_2\|_F^2\,.$$

By Lemma 2.1,

$$\sigma_n^2(\widetilde{U}_1^* U_1) \sum_{i=1}^{r} |\widetilde{\lambda}_i - \lambda_i|^2 \le \|\widetilde{\Lambda}_1 \widetilde{U}_1^* U_1 - \widetilde{U}_1^* U_1 \Lambda_1\|_F^2 \, . \tag{2.6}$$

By the C-S decomposition theorem (see, e.g., [11]), we have

$$\sigma_n^2(\widetilde{U}_1^* U_1) = 1 - \|\sin\Theta(U_1, \widetilde{U}_1)\|_2^2,$$

and therefore,

$$\delta_{12}^2 \|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 + (1 - \|\sin\Theta(U_1, \widetilde{U}_1)\|_2^2) \sum_{i=1}^{r} |\widetilde{\lambda}_i - \lambda_i| \le ||R||_F^2,$$

which proves (2.3). □

Obviously the combined bounds in Theorem 2.2 contain perturbation bounds for both eigenspaces and eigenvalues. If we take $U_1 = U$ and $\widetilde{U}_1 = \widetilde{U}$, then $\|\sin\Theta(U_1, \widetilde{U}_1)\|_2 = \|\sin\Theta(U_1, \widetilde{U}_1)\|_F = 0$ and the bound (2.3) reduces to the Hoffman–Wielandt theorem. It is easy to obtain Davis and Kahan's sinΘ theorem from the bound (2.3) since $\|\sin\Theta(U_1, \widetilde{U}_1)\|_2 \le 1$.

[illegible] 2.1. Let

$$H = U\Lambda U^*, \qquad \widetilde{H} = (1+\epsilon) U \Lambda U^*,$$

where $\Lambda$ is positive and diagonal and

$$\Lambda = \begin{pmatrix} I_r & 0 \\ 0 & 2I_{n-r} \end{pmatrix}, \qquad U = \begin{pmatrix} U_{11} & 0 \\ 0 & U_{22} \end{pmatrix}.$$

Then

$$\Delta H = \epsilon U \begin{pmatrix} I_r & 0 \\ 0 & 2I_{n-r} \end{pmatrix} U^*, \qquad \Delta H U_1 = \epsilon \begin{pmatrix} U_{11} \\ 0 \end{pmatrix}.$$

A simple calculation gives

$$\|R\|_F^2 = \|\Delta H U_1\|_F^2 = r\epsilon^2, \qquad \delta_{12} = 1 + 2\epsilon, \qquad \delta_{11} = \epsilon.$$

The bound (1.4) becomes

$$\|\sin\Theta(U_1 \, \widetilde{U}_1)\|_F^2 \le \frac{\|R\|_F^2}{\delta_{12}^2} = \frac{r\epsilon^2}{(1+2\epsilon)^2},$$

and from our bound (2.2),

$$\|\sin\Theta(U_1 \, \widetilde{U}_1)\|_F^2 \le \frac{\|R\|_F^2 - r\delta_{11}^2}{\delta_{12}^2 - \delta_{11}^2} = 0\,,$$

which leads to $\|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 = 0$.

When $\delta_{12} > 0$,

$$\|\sin\Theta(U_1, \widetilde{U}_1)\|_F \le O(\|R\|_F), \qquad \sum_{i=1}^{r} (\lambda_i - \widetilde{\lambda}_i)^2 \le \|R\|_F^2$$

and we obtain the asymptotic bound in (1.8). However, the following example shows the absolute bound

$$\delta_{12}^2 \|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 + \sum_{i=1}^{r} (\lambda_i - \widetilde{\lambda}_i)^2 \le \|R\|_F^2$$

does not hold.

[illegible] 2.2. Let

$$H = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \widetilde{H} = (1+\epsilon)\widetilde{U}^T H \widetilde{U},$$

where $\epsilon > 0$ and

$$\widetilde{U} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The eigenvalues of $H$ and $\widetilde{H}$ are $2, 1, 1$ and $2(1+\epsilon), (1+\epsilon), (1+\epsilon)$, respectively. For $r = 1$,

$$U_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \widetilde{U}_1 = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \end{pmatrix}.$$

A simple calculation gives $\delta_{12} = 1 - \epsilon$ and

$$\begin{aligned} \delta_{12}^2 \|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 + \sum_{i=1}^{r} |\widetilde{\lambda}_i - \lambda_i|^2 &= (1-\epsilon)^2 \sin^2\theta + (2\epsilon)^2 \\ &> (1-\epsilon)^2 \sin^2\theta + 4\epsilon^2 \cos^2\theta = ||\Delta H U_1||_F^2 = \|R\|_F^2 . \end{aligned}$$

The perturbation bounds for eigenspaces in a relative measure have been studied by several authors; Dopico, Moro, and Molera [4] presented the relative perturbation bound

$$\delta_{12}^{(\frac{1}{2},\frac{1}{2})} \|\sin\Theta(U_1\, \widetilde{U}_1)\|_F \le \|H^{-1/2} \Delta H \widetilde{H}^{-1/2}\|_F$$

for nonsingular Hermitian matrices $H$ and $\widetilde{H}$. A sharper bound obtained by Chen and Li [2] is

$$\frac{2\delta_{12}^{(\frac{1}{2},\frac{1}{2})} \delta_{21}^{(\frac{1}{2},\frac{1}{2})}}{\sqrt{\left(\delta_{12}^{(\frac{1}{2},\frac{1}{2})}\right)^2 + \left(\delta_{21}^{(\frac{1}{2},\frac{1}{2})}\right)^2}} \|\sin\Theta(U_1\, \widetilde{U}_1)\|_F \le \|H^{-1/2} \Delta H \widetilde{H}^{-1/2}\|_F . \tag{2.7}$$

Li [8] and Londre and Rhee [9] studied perturbation bounds in a different relative measure. The perturbation bound in [8, 9] is given by

$$\delta_{12}^{(\frac{1}{2},\frac{1}{2})} \|\sin\Theta(U_1\, \widetilde{U}_1)\|_F \le \frac{\|H^{-1/2} \Delta H H^{-1/2}\|_F}{\sqrt{1-\mu_2}}, \tag{2.8}$$

where

$$\mu_2 = \|H^{-1/2} \Delta H H^{-1/2}\|_2 .$$

A modified bound in the relative measure given in [2] is

$$\min\left\{\delta_{12}^{(\frac{1}{2},\frac{1}{2})}, \delta_{21}^{(\frac{1}{2},\frac{1}{2})}\right\} \|\sin\Theta(U_1,\widetilde{U}_1)\|_F \le \frac{\sqrt{2}}{2}\frac{\|H^{-1/2}\Delta H H^{-1/2}\|_F}{\sqrt{1-\mu_2}}. \tag{2.9}$$

The perturbation bound of eigenvalues in the more general relative measure $\|H^{-k}\Delta H\widetilde{H}^{-l}\|_F$ with any nonnegative numbers $k$ and $l$ was studied by Ipsen [7]. The perturbation bound given in [7] is

$$\sum_{i=1}^{n} |\lambda_i^{-k}\widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_{\tau(i)}^{-l}|^2 \le \|H^{-k}\Delta H\widetilde{H}^{-l}\|_F^2. \tag{2.10}$$

No perturbation bound for eigenspaces has been obtained.

Now we extend our analysis for combined perturbation bounds to these relative measures, instead of the measure $\|R\|_F$ used in Theorem 2.2. Since

$$H^{-k}\Delta H\widetilde{H}^{-l} = H^{-k}\widetilde{H}^{1-l} - H^{1-k}\widetilde{H}^{-l},$$

multiplying on the left by $U^*$ and the right by $\widetilde{U}$ gives

$$\Lambda^{-k}U^*\widetilde{U}\widetilde{\Lambda}^{1-l} - \Lambda^{1-k}U^*\widetilde{U}\widetilde{\Lambda}^{-l} = U^*H^{-k}\Delta H\widetilde{H}^{-l}\widetilde{U},$$

which can be rewritten in the block form as

$$\begin{pmatrix} \Lambda_1^{-k}U_1^*\widetilde{U}_1\widetilde{\Lambda}_1^{1-l} - \Lambda_1^{1-k}U_1^*\widetilde{U}_1\widetilde{\Lambda}_1^{-l} & \Lambda_1^{-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{1-l} - \Lambda_1^{1-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{-l} \\ \Lambda_2^{-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{1-l} - \Lambda_2^{1-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{-l} & \Lambda_2^{-k}U_2^*\widetilde{U}_2\widetilde{\Lambda}_2^{1-l} - \Lambda_2^{1-k}U_2^*\widetilde{U}_2\widetilde{\Lambda}_2^{-l} \end{pmatrix} = U^*H^{-k}\Delta H\widetilde{H}^{-l}\widetilde{U}.$$

It follows that

$$\begin{aligned} &\|\Lambda_1^{-k}U_1^*\widetilde{U}_1\widetilde{\Lambda}_1^{1-l} - \Lambda_1^{1-k}U_1^*\widetilde{U}_1\widetilde{\Lambda}_1^{-l}\|_F^2 + \|\Lambda_1^{-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{1-l} - \Lambda_1^{1-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{-l}\|_F^2 \\ &+\|\Lambda_2^{-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{1-l} - \Lambda_2^{1-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{-l}\|_F^2 + \|\Lambda_2^{-k}U_2^*\widetilde{U}_2\widetilde{\Lambda}_2^{1-l} - \Lambda_2^{1-k}U_2^*\widetilde{U}_2\widetilde{\Lambda}_2^{-l}\|_F^2 \\ &= \|U^*H^{-k}\Delta H\widetilde{H}^{-l}\widetilde{U}\|_F^2. \end{aligned} \tag{2.11}$$

We take the same approach as used for (2.5). Since

$$\begin{aligned} &|(\Lambda_1^{-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{1-l} - \Lambda_1^{1-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{-l})_{ij}|^2 = (\lambda_i^{-k}\widetilde{\lambda}_j^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_j^{-l})^2|(U_1^*\widetilde{U}_2)_{ij}|^2 \\ &\ge \left(\delta_{12}^{(k,l)}\right)^2 |(U_1^*\widetilde{U}_2)_{ij}|^2 \end{aligned}$$

and

$$\begin{aligned} &|(\Lambda_2^{-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{1-l} - \Lambda_2^{1-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{-l})_{ij}|^2 = (\lambda_j^{-k}\widetilde{\lambda}_i^{1-l} - \lambda_j^{1-k}\widetilde{\lambda}_i^{-l})^2|(U_2^*\widetilde{U}_1)_{ij}|^2 \\ &\ge \left(\delta_{21}^{(k,l)}\right)^2 |(U_2^*\widetilde{U}_1)_{ij}|^2, \end{aligned}$$

we obtain

$$\begin{aligned} &\left(\delta_{12}^{(k,l)}\right)^2 \|U_1^*\widetilde{U}_2\|_F^2 \le \|\Lambda_1^{-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{1-l} - \Lambda_1^{1-k}U_1^*\widetilde{U}_2\widetilde{\Lambda}_2^{-l}\|_F^2, \\ &\left(\delta_{21}^{(k,l)}\right)^2 \|U_2^*\widetilde{U}_1\|_F^2 \le \|\Lambda_2^{-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{1-l} - \Lambda_2^{1-k}U_2^*\widetilde{U}_1\widetilde{\Lambda}_1^{-l}\|_F^2. \end{aligned} \tag{2.12}$$

On the other hand, let

$$T = \begin{pmatrix} U_1^* \widetilde{U}_1 & \\ & U_2^* \widetilde{U}_2 \end{pmatrix}, \quad D_1 = \begin{pmatrix} \Lambda_1^{-k} & \\ & \Lambda_2^{-k} \end{pmatrix}, \quad D_3 = \begin{pmatrix} \Lambda_1^{1-k} & \\ & \Lambda_2^{1-k} \end{pmatrix}$$

and

$$D_2 = \begin{pmatrix} \widetilde{\Lambda}_1^{1-l} & \\ & \Lambda_2^{1-l} \end{pmatrix}, \quad D_4 = \begin{pmatrix} -\widetilde{\Lambda}_1^{-l} & \\ & -\widetilde{\Lambda}_2^{-l} \end{pmatrix}.$$

We have

$$\begin{aligned} &\|\Lambda_1^{-k} U_1^* \widetilde{U}_1 \widetilde{\Lambda}_1^{1-l} - \Lambda_1^{1-k} U_1^* \widetilde{U}_1 \widetilde{\Lambda}_1^{-l}\|_F^2 + \|\Lambda_2^{-k} U_2^* \widetilde{U}_2 \widetilde{\Lambda}_2^{1-l} - \Lambda_2^{1-k} U_2^* \widetilde{U}_2 \widetilde{\Lambda}_2^{-l}\|_F^2 \\ &= \|D_1 T D_2 - D_3 T D_4\|_F^2. \end{aligned} \tag{2.13}$$

By Lemma 2.1, there exists a permutation $\tau$ of $\langle n \rangle$ such that

$$\|D_1 T D_2 - D_3 T D_4\|_F^2 \geq \sigma_n^2(T) \sum_{i=1}^{n} |\lambda_i^{-k} \widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k} \widetilde{\lambda}_{\tau(i)}^{-l}|^2. \tag{2.14}$$

By the C-S decomposition theorem [11], it is easy to see that

$$\sigma_n^2(T) \geq \min\{\sigma_n^2(U_1^* \widetilde{U}_1),\ \sigma_n^2(U_2^* \widetilde{U}_2)\} = 1 - \|\sin\Theta(U_1, \widetilde{U}_1)\|_2^2.$$

It follows that

$$\begin{aligned} &(1 - \|\sin\Theta(U_1, \widetilde{U}_1)\|_2^2) \sum_{i=1}^{r} |\lambda_i^{-k} \widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k} \widetilde{\lambda}_{\tau(i)}^{-l}|^2 \\ &\leq \|\Lambda_1^{-k} U_1^* \widetilde{U}_1 \widetilde{\Lambda}_1^{1-l} - \Lambda_1^{1-k} U_1^* \widetilde{U}_1 \widetilde{\Lambda}_1^{-l}\|_F^2 + \|\Lambda_2^{-k} U_2^* \widetilde{U}_2 \widetilde{\Lambda}_2^{1-l} - \Lambda_2^{1-k} U_2^* \widetilde{U}_2 \widetilde{\Lambda}_2^{-l}\|_F^2. \end{aligned} \tag{2.15}$$

Substituting (2.12) and (2.15) into (2.11) leads to a combined bound in the following theorem.

THEOREM 2.3. [illegible] $H$ [illegible] $\widetilde{H} = H + \Delta H$ [illegible] $n \times n$ [illegible] (1.1)–(1.2) [illegible]

$$\begin{aligned} &\left((\delta_{12}^{(k,l)})^2 + (\delta_{21}^{(k,l)})^2\right) \|\sin\Theta(U_1, \widetilde{U}_1)\|_F^2 \\ &\qquad + (1 - \|\sin\Theta(U_1, \widetilde{U}_1)\|_2^2) \sum_{i=1}^{n} (\lambda_i^{-k} \widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k} \widetilde{\lambda}_{\tau(i)}^{-l})^2 \\ &\qquad \leq ||H^{-k} \Delta H \widetilde{H}^{-l}||_F^2. \end{aligned} \tag{2.16}$$

By an analogous approach, we can obtain

$$\begin{aligned} \left(\delta_{22}^{(k,l)}\right)^2 \|U_2^* \widetilde{U}_2\|_F^2 &\leq \|\Lambda_2^{-k} U_2^* \widetilde{U}_2 \widetilde{\Lambda}_2^{1-l} - \Lambda_2^{1-k} U_2^* \widetilde{U}_2 \widetilde{\Lambda}_2^{-l}\|_F^2, \\ \left(\delta_{11}^{(k,l)}\right)^2 \|U_1^* \widetilde{U}_1\|_F^2 &\leq \|\Lambda_1^{-k} U_1^* \widetilde{U}_1 \widetilde{\Lambda}_1^{1-l} - \Lambda_1^{1-k} U_1^* \widetilde{U}_1 \widetilde{\Lambda}_1^{-l}\|_F^2. \end{aligned} \tag{2.17}$$

It is easy to see that

$$\|\sin\Theta(U_1, \widetilde{U}_1)\|_F = \|U_1^* \widetilde{U}_2\|_F = \|U_2^* \widetilde{U}_1\|_F.$$

By the definition of $\cos\Theta$,

$$\|\cos\Theta(U_1,\widetilde{U}_1)\|_F = \|U_1^*\widetilde{U}_1\|_F\,, \qquad \|\cos\Theta(U_2,\widetilde{U}_2)\|_F = \|U_2^*\widetilde{U}_2\|_F,$$

and again by the C-S decomposition theorem [11], we have

$$\|\cos\Theta(U_2,\widetilde{U}_2)\|_F^2 = \|\cos\Theta(U_1,\widetilde{U}_1)\|_F^2 + n - 2r$$

and

$$\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 = r - \|\cos\Theta(U_1,\widetilde{U}_1)\|_F^2\,.$$

A new bound is given in the following theorem. The proof can be obtained by following the proof of Theorem 2.3 and replacing (2.15) by (2.17).

THEOREM 2.4. [illegible] $H$ [illegible] $\widetilde{H} = H + \Delta H$ [illegible] $n \times n$ [illegible] (1.1)–(1.2) [illegible]

$$\begin{aligned}&\Big((\delta_{12}^{(k,l)})^2 + (\delta_{21}^{(k,l)})^2 - (\delta_{11}^{(k,l)})^2 - (\delta_{22}^{(k,l)})^2\Big)\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2\\ &\quad + r(\delta_{11}^{(k,l)})^2 + (n-r)(\delta_{22}^{(k,l)})^2\\ &\quad \le ||H^{-k}\Delta H\widetilde{H}^{-l}||_F^2.\end{aligned} \tag{2.18}$$

[illegible] 2.1. Let $H$ and $\widetilde{H}$ be Hermitian and the eigenvalues of $H$ and $\widetilde{H}$ be enumerated by

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n \quad \text{and} \quad \widetilde{\lambda}_1 \le \widetilde{\lambda}_2 \le \cdots \le \widetilde{\lambda}_n,$$

and assume that

$$\lambda_r < \lambda_{r+1}, \qquad \widetilde{\lambda}_r < \widetilde{\lambda}_{r+1}.$$

When the perturbation is small enough, we always have

$$\sum_{i=1}^n |\lambda_i^{-k}\widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_{\tau(i)}^{-l}|^2 \le \sqrt{(\delta_{12}^{(k,l)})^2 + (\delta_{21}^{(k,l)})^2}$$

for some permutation $\tau$ of $\langle n\rangle$ and, therefore,

$$\begin{aligned}&\Big((\delta_{12}^{(k,l)})^2 + (\delta_{21}^{(k,l)})^2\Big)\,\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 + \|\cos\Theta(U_1,\widetilde{U}_1)\|_2^2 \sum_{i=1}^n |\lambda_i^{-k}\widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_{\tau(i)}^{-l}|^2\\ &\ge \Big(\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 + 1 - \|\sin\Theta(U_1,\widetilde{U}_1)\|_2^2\Big)\sum_{i=1}^n |\lambda_i^{-k}\widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_{\tau(i)}^{-l}|^2\\ &\ge \sum_{i=1}^n |\lambda_i^{-k}\widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_{\tau(i)}^{-l}|^2,\end{aligned}$$

which implies that the bound (2.16) is strictly sharper than the bound in (2.10) for the eigenvalue perturbation. Since $\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 = \min\{r\,, n-r\}$, we have the following corollary.

COROLLARY 2.5. [illegible] 2.4

$$({\delta_{12}^{(k,l)}}^2 + {\delta_{21}^{(k,l)}}^2)\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 \le \begin{cases} ||H^{-k}\Delta H\widetilde{H}^{-l}||_F^2 - (n-2r){\delta_{22}^{(k,l)}}^2, & 2r \le n,\\ ||H^{-k}\Delta H\widetilde{H}^{-l}||_F^2 - (2r-n){\delta_{11}^{(k,l)}}^2, & 2r > n.\end{cases} \tag{2.19}$$

In comparison with the perturbation bound in (2.7), our bound (2.19) is sharper. The following corollary gives two combined perturbation bounds, in terms of the relative measure $\|H^{-k}\Delta H H^{-l}\|_F^2/(1-\mu_2)^{2l}$, which are sharper than the corresponding bounds obtained in [2, 8, 9].

Corollary 2.6. $\mu_2 = \left\|H^{-1/2}\Delta H H^{-1/2}\right\|_2 < 1$

$$\begin{aligned}\Big((\delta_{12}^{(k,l)})^2 + (\delta_{21}^{(k,l)})^2 - (\delta_{11}^{(k,l)})^2 - (\delta_{22}^{(k,l)})^2\Big)\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 \\ + r(\delta_{11}^{(k,l)})^2 + (n-r)(\delta_{22}^{(k,l)})^2 \\ \le \frac{\left\|H^{-k}\Delta H H^{-l}\right\|_2^2}{(1-\mu_2)^{2l}}\end{aligned} \tag{2.20}$$

$$\begin{aligned}\Big((\delta_{12}^{(k,l)})^2 + (\delta_{21}^{(k,l)})^2\Big)\|\sin\Theta(U_1,\widetilde{U}_1)\|_F^2 \\ + (1 - \|\sin\Theta(U_1,\widetilde{U}_1)\|_2^2)\sum_{i=1}^{n}|\lambda_i^{-k}\widetilde{\lambda}_{\tau(i)}^{1-l} - \lambda_i^{1-k}\widetilde{\lambda}_{\tau(i)}^{-l}|^2 \\ \le \frac{\left\|H^{-k}\Delta H H^{-l}\right\|_2^2}{(1-\mu_2)^{2l}}.\end{aligned} \tag{2.21}$$

Taking the same approach as in [2], one may deduce that

$$||H^{-k}\Delta H\widetilde{H}^{-l}||_F \le \frac{\left\|H^{-k}\Delta H H^{-l}\right\|_2}{(1-\mu_2)^l}, \tag{2.22}$$

which together with Theorems 2.2 and 2.3 gives the desired bound. $\square$

**3. Combined bounds for singular value decompositions.** Let $A$, $\widetilde{A} \in C^{m\times n}$ have the singular value decompositions (SVDs)

$$A = U\Sigma V^* = \begin{pmatrix} U_1 & U_2 \end{pmatrix}\begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}\begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix} \tag{3.1}$$

and

$$\widetilde{A} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^* = \begin{pmatrix} \widetilde{U}_1 & \widetilde{U}_2 \end{pmatrix}\begin{pmatrix} \widetilde{\Sigma}_1 & 0 \\ 0 & \widetilde{\Sigma}_2 \end{pmatrix}\begin{pmatrix} \widetilde{V}_1^* \\ \widetilde{V}_2^* \end{pmatrix}, \tag{3.2}$$

where $U = (\widetilde{U}_1\ \widetilde{U}_2)$ and $\widetilde{U} = (\widetilde{U}_1\ \widetilde{U}_2)$ are $m\times m$ unitary, $V = (\widetilde{V}_1\ \widetilde{V}_2)$ and $\widetilde{V} = (\widetilde{V}_1\ \widetilde{V}_2)$ are $n \times n$ unitary, and

$$\Sigma_1 = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r), \quad \Sigma_2 = \operatorname{diag}(\sigma_{r+1}, \sigma_{r+2}, \ldots, \sigma_n), \tag{3.3}$$

$$\widetilde{\Sigma}_1 = \operatorname{diag}(\widetilde{\sigma}_1, \widetilde{\sigma}_2, \ldots, \widetilde{\sigma}_r), \quad \widetilde{\Sigma}_2 = \operatorname{diag}(\widetilde{\sigma}_{r+1}, \widetilde{\sigma}_{r+2}, \ldots, \widetilde{\sigma}_n). \tag{3.4}$$

Let $\sigma(A) = \lambda(\sqrt{A^*A})$ be the set of singular values of $A$ and

$$\sigma_{ext}(\Sigma_2) = \begin{cases} \sigma(\Sigma_2) \cup \{0\} & \text{if } m > n, \\ \sigma(\Sigma_2) & \text{if } m = n. \end{cases}$$

The perturbation of singular subspaces is usually measured by the angle between the subspaces $\Re(U_1)$ and $\Re(\widetilde{U}_1)$ and the angle between the subspaces $\Re(V_1)$ and $\Re(\widetilde{V}_1)$, denoted by

$$\|\sin\Theta\|_F = \|U_1^*\widetilde{U}_2\|_F, \qquad \|\sin\Phi\|_F = \|V_1^*\widetilde{V}_2\|_F,$$

respectively. Let

$$\epsilon_{ij}^{(k,l)} = \min_{\lambda\in\sigma_{ext}(\Sigma_i),\widetilde{\mu}\in\sigma(\widetilde{\Sigma}_j)} \frac{|\mu-\widetilde{\mu}|}{|\mu|^k|\widetilde{\mu}|^l}, \quad i,j=1,2.$$

Similarly we use the notation $\epsilon_{ij} = \epsilon_{ij}^{(0,0)}$.

The perturbation bound of singular subspace was given by Wedin [13] and the perturbation bound for singular values can be found in the literature. We summarize the results in the following theorem.

THEOREM B. $A$ $\widetilde{A} \in C^{m\times n}$ (3.1)–(3.4).

$$\epsilon_{12}^2(\|\sin\Theta\|_F^2 + \|\sin\Phi\|_F^2) \le \|R\|_F^2 + \|S\|_F^2 \tag{3.5}$$

$$2\sum_{i=1}^{r}(\sigma_i - \widetilde{\sigma}_{\tau(i)})^2 \le \|R\|_F^2 + \|S\|_F^2, \tag{3.6}$$

$$R = A\widetilde{V}_1 - \widetilde{U}_1\widetilde{\Sigma}_1 = -E\widetilde{V}_1, \qquad S = A^*\widetilde{U}_1 - \widetilde{V}_1\widetilde{\Sigma}_1 = -E^*\widetilde{U}_1.$$

To obtain a combined perturbation bound for SVDs, we consider the Jordan–Wielandt matrices

$$\mathbb{H} = \begin{pmatrix} 0 & A^* \\ A & 0 \end{pmatrix} \quad\text{and}\quad \widetilde{\mathbb{H}} = \begin{pmatrix} 0 & \widetilde{A}^* \\ \widetilde{A} & 0 \end{pmatrix}. \tag{3.7}$$

Let

$$\mathbb{U} = \frac{1}{\sqrt{2}}\begin{pmatrix} V_1 & V_1 & V_2 & V_2 \\ U_1 & -U_1 & U_2 & -U_2 \end{pmatrix} = \begin{pmatrix} \mathbb{U}_1 & \mathbb{U}_2 \end{pmatrix}$$

and

$$\widetilde{\mathbb{U}} = \frac{1}{\sqrt{2}}\begin{pmatrix} \widetilde{V}_1 & \widetilde{V}_1 & \widetilde{V}_2 & \widetilde{V}_2 \\ \widetilde{U}_1 & -\widetilde{U}_1 & \widetilde{U}_2 & -\widetilde{U}_2 \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbb{U}}_1 & \widetilde{\mathbb{U}}_2 \end{pmatrix},$$

where

$$\mathbb{U}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} V_1 & V_1 \\ U_1 & -U_1 \end{pmatrix} \quad\text{and}\quad \widetilde{\mathbb{U}}_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} \widetilde{V}_1 & \widetilde{V}_1 \\ \widetilde{U}_1 & -\widetilde{U}_1 \end{pmatrix}.$$

Then the eigendecomposition of $\mathbb{H}$ and $\widetilde{\mathbb{H}}$ can be rewritten as

$$\mathbb{H} = \mathbb{U}\begin{pmatrix} \Sigma_1 & & & \\ & -\Sigma_1 & & \\ & & \Sigma_2 & \\ & & & -\Sigma_2 \end{pmatrix}\mathbb{U}^* \quad\text{and}\quad \widetilde{\mathbb{H}} = \widetilde{\mathbb{U}}\begin{pmatrix} \widetilde{\Sigma}_1 & & & \\ & -\widetilde{\Sigma}_1 & & \\ & & \widetilde{\Sigma}_2 & \\ & & & -\widetilde{\Sigma}_2 \end{pmatrix}\widetilde{\mathbb{U}}^*,$$

respectively. Applying Theorems 2.2 and 2.3 to the matrices $\mathbb{H}$ and $\widetilde{\mathbb{H}}$, we have the following combined perturbation bounds for singular values and singular subspaces.

THEOREM 3.1. $A$ $\widetilde{A} = A + E$ $n \times n$ (3.1)–(3.4) $\tau$ $\langle n \rangle$

$$\epsilon_{12}^2 \left(\|\sin\Phi(U_1, \widetilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \widetilde{V}_1\right)\|_F^2) + \left(2 - \|\sin\Phi(U_1, \widetilde{U}_1)\|_2^2 - \|\sin\Theta(V_1, \widetilde{V}_1)\|_2^2\right)\sum_{i=1}^{r}(\sigma_i - \widetilde{\sigma}_{\tau(i)})^2 \le \|R\|_F^2 + \|S\|_F^2 \tag{3.8}$$

$$(\epsilon_{12}^2 - \epsilon_{11}^2)(\|\sin\Phi(U_1, \widetilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \widetilde{V}_1)\|_F^2) + 2r\epsilon_{11}^2 \le \|R\|_F^2 + \|S\|_F^2. \tag{3.9}$$

In an asymptotic sense, (3.9) becomes

$$\begin{aligned}&\epsilon_{12}^2 \left(\|\sin\Phi(U_1, \widetilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \widetilde{V}_1)\|_F^2\right) + 2\sum_{i=1}^{n}|\sigma_i - \widetilde{\sigma}_i|^2 \\ &\quad \le (\|R\|_F^2 + \|S\|_F^2) + O((\|R\|_F^2 + \|S\|_F^2)^2).\end{aligned} \tag{3.10}$$

From the SVDs (3.1) and (3.2), we obtain the left polar decomposition of the matrices $A$ and $\widetilde{A}$, defined by

$$A = QP_l \quad \text{and} \quad \widetilde{A} = \widetilde{Q}\widetilde{P}_l, \tag{3.11}$$

and, similarly, the right polar decomposition

$$A = P_rQ \quad \text{and} \quad \widetilde{A} = \widetilde{P}_r\widetilde{Q}, \tag{3.12}$$

where $Q$ is called the unitary polar factor of $A$ and $P_l$ and $P_r$ are called the left and right Hermitian factor, respectively. It is noted that Wedin's $\sin\theta$ theorem is given in an absolute measure $\|R\|_F^2 + \|S\|_F^2$. The perturbation bounds for singular values and singular subspaces in some relative measures was studied in [4], where the relative measures $\|P_l^{-k}E\widetilde{P}_r^{-l}\|_F$ and $\|\widetilde{P}_l^{-l}EP_r^{-k}\|_F$ are used. The extension to combined bounds is given in the following theorem and the proof is similar to the proofs for Theorems 2.3 and 2.4.

THEOREM 3.2. $A$ $\widetilde{A} = A + E$ $n \times n$ (3.1)–(3.4)

$$\begin{aligned}&\left(\left(\epsilon_{21}^{(k,l)}\right)^2 + \left(\epsilon_{12}^{(k,l)}\right)^2\right)\left(\|\sin\Phi(U_1, \widetilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \widetilde{V}_1)\|_F^2\right) \\ &\quad + \left(2 - \|\sin\Phi(U_1, \widetilde{U}_1)\|_2^2 - \|\sin\Theta(V_1, \widetilde{V}_1)\|_2^2\right)\sum_{i=1}^{n}(\sigma_i - \widetilde{\sigma}_{\tau(i)})^2 \\ &\quad \le \|P_l^{-k}E\widetilde{P}_r^{-l}\|_F^2 + \|\widetilde{P}_l^{-l}EP_r^{-k}\|_F^2\end{aligned} \tag{3.13}$$

$$\begin{aligned}&\left(\left(\epsilon_{21}^{(k,l)}\right)^2 + \left(\epsilon_{12}^{(k,l)}\right)^2 - \left(\epsilon_{11}^{(k,l)}\right)^2 - \left(\epsilon_{22}^{(k,l)}\right)^2\right)\left(\|\sin\Phi(U_1, \widetilde{U}_1)\|_F^2 + \|\sin\Theta(V_1, \widetilde{V}_1)\|_F^2\right) \\ &+ 2r\left(\epsilon_{11}^{(k,l)}\right)^2 + 2(n-r)\left(\epsilon_{22}^{(k,l)}\right)^2 \le \|P_l^{-k}E\widetilde{P}_r^{-l}\|_F^2 + \|\widetilde{P}_l^{-l}EP_r^{-k}\|_F^2,\end{aligned} \tag{3.14}$$

$P_l$ $P_r$ $\widetilde{P}_l$ $\widetilde{P}_r$ (3.11) (3.12)
3.1. A perturbation bound in Theorem 3.4 of [4] is as follows:

$$\left(\epsilon_{21}^{(\frac{1}{2},\frac{1}{2})}\right)^2 \left(\|\sin\Phi(U_1,\widetilde{U}_1)\|_F^2 + \|\sin\Theta(V_1,\ \widetilde{V}_1)\|_F^2\right)$$
$$\leq \|P_l^{-\frac{1}{2}} E \widetilde{P}_r^{-\frac{1}{2}}\|_F^2 + \|\widetilde{P}_l^{-\frac{1}{2}} E P_r^{-\frac{1}{2}}\|_F^2,$$

which can also be obtained from Theorem 3.2.

Finally, we consider the bounds for the right singular subspaces as in [9]. Let

$$\delta \equiv ||EA^\dagger||_2, \quad \delta_F \equiv ||EA^\dagger||_F.$$

Let $A$ and $\widetilde{A}$ have the SVDs (3.1)–(3.4) and let

$$\varsigma_{ij} = \min_{\mu\in\sigma(\Sigma_i),\widetilde{\mu}\in\sigma(\widetilde{\Sigma}_j)} \frac{|\widetilde{\mu}^2-\mu^2|}{\widetilde{\mu}\mu}.$$

Let $H = A^*A$ and $\widetilde{H} = \widetilde{A}^*\widetilde{A}$. Then $H$ and $\widetilde{H}$ have the eigendecompositions

$$H = (V_1, V_2)\begin{pmatrix} \Sigma_1^2 & 0 \\ 0 & \Sigma_2^2 \end{pmatrix}\begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix}, \qquad \widetilde{H} = \begin{pmatrix} \widetilde{V}_1 & \widetilde{V}_2 \end{pmatrix}\begin{pmatrix} \widetilde{\Sigma}_1^2 & 0 \\ 0 & \widetilde{\Sigma}_2^2 \end{pmatrix}\begin{pmatrix} \widetilde{V_1}^* \\ \widetilde{V_2}^* \end{pmatrix}.$$

Applying Corollary 2.6 to $H$ and $\widetilde{H}$ with $l = k = 1/2$ by the same argument as in Theorem 2.1 of [9], we obtain the following estimate:

$$(\varsigma_{21}^2 + \varsigma_{12}^2 - \varsigma_{11}^2 - \varsigma_{22}^2)||\sin\Theta(V_1,\widetilde{V}_1)||_F^2 + r\varsigma_{11}^2 + (n-r)\varsigma_{22}^2 \leq \frac{(2\delta_F+\delta_F^2)^2}{1-3\delta} \tag{3.15}$$

for $\delta < 1/3$. A simpler form of (3.15) is

$$||\sin\Theta(V_1,\widetilde{V}_1)||_F \leq \frac{2\delta_F+\delta_F^2}{\sqrt{(1-3\delta)(\varsigma_{21}^2+\varsigma_{12}^2)}}. \tag{3.16}$$

It is easy to see that the bound in (3.16) is always sharper than those in Theorem 2.1 of [9].

**Acknowledgment.** The authors would like to thank the referee for valuable comments.

## REFERENCES

[1] J. L. Barlow and I. Slapnicar, *Optimal perturbation bounds for the Hermitian eigenvalue problem*, Linear Algebra Appl., 309 (2000), pp. 19–43.

[2] X. Chen and W. Li, *A note on the perturbation bounds of eigenspaces for Hermitian matrices*, J. Comput. Appl. Math., 196 (2006), pp. 338–346.

[3] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation.* III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

[4] F. M. Dopico, J. Moro, and J. M. Molera, *Weyl-type relative perturbation bounds for eigensystems of Hermitian matrices*, Linear Algebra Appl., 309 (2000), pp. 3–18.

[5] L. Elsner and S. Friedland, *Singular values, doubly stochastic matrices, and applications*, Linear Algebra Appl., 220 (1995), pp. 161–169.

[6] A. J. Hoffman and H. W. Wielandt, *The variation of spectrum of a normal matrix*, Duke Math. J., 20 (1953), pp. 37–39.

[7] I. C. F. Ipsen, *A note on unifying absolute and relative perturbation bounds*, Linear Algebra Appl., 358 (2003), pp. 239–253.

[8] R.-C. Li, *Relative perturbation theory.* II. *Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 471–492.

[9] T. Londre and N. H. Rhee, *A note on relative perturbation bounds*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 357–361.

[10] R. Mathias and K. Veselic, *A relative perturbation bound for positive definite matrices*, Linear Algebra Appl., 270 (1998), pp. 315–321.

[11] G. Stewart and J. Sun, *Matrix Perturbation Theory*, Academic Press, Boston, 1990.

[12] N. Truhar and I. Slapnicar, *Relative perturbation bound for invariant subspaces of graded indefinite Hermitian matrices*, Linear Algebra Appl., 301 (1999), pp. 171–185.

[13] P. A. Wedin, *Perturbation bounds in connection with the singular value decomposition*, BIT, 12 (1972), pp. 99–111.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 656–659

# GENERALIZED RANK-CONSTRAINED MATRIX APPROXIMATIONS*

SHMUEL FRIEDLAND[†] AND ANATOLI TOROKHTI[‡]

**Abstract.** In this paper we give an explicit solution to the rank-constrained matrix approximation in Frobenius norm, which is a generalization of the classical approximation of an $m \times n$ matrix $A$ by a matrix of, at most, rank $k$.

**Key words.** singular value decomposition (SVD), generalized rank-constrained matrix approximations, generalized inverse

**AMS subject classification.** 15A18

**DOI.** 10.1137/06065551

## 1. Introduction.

Let $\mathbb{C}^{m\times n}$ be a set of $m \times n$ complex valued matrices, and denote by $\mathcal{R}(m,n,k) \subseteq \mathbb{C}^{m\times n}$ the variety of all $m \times n$ matrices of, at most, rank $k$. Fix $A = [a_{ij}]_{i,j=1}^{m,n} \in \mathbb{C}^{m\times n}$; then $A^* \in \mathbb{C}^{n\times m}$ is the conjugate transpose of $A$ and $||A||_F := \sqrt{\sum_{i,j=1}^{m,n} |a_{ij}|^2}$ is the Frobenius norm of $A$. Recall that the *singular value decomposition* of $A$, abbreviated here as *SVD*, is given by $A = U_A \Sigma_A V_A^*$, where $U_A \in \mathbb{C}^{m\times m}, V_A \in \mathbb{C}^{n\times n}$ are unitary matrices and $\Sigma_A := \operatorname{diag}(\sigma_1(A), \ldots, \sigma_{\min(m,n)}(A)) \in \mathbb{C}^{m\times n}$ is a generalized diagonal matrix, with the singular values $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq 0$ on the main diagonal. The number of positive singular values of $A$ is $r$, which is equal to the rank of $A$, denoted by $\operatorname{rank} A$. Let $U_A = [\mathbf{u}_1,\ \mathbf{u}_2, \ldots, \mathbf{u}_m], V_A = [\mathbf{v}_1,\ \mathbf{v}_2, \ldots, \mathbf{v}_n]$ be the representations of $U, V$ in terms of their $m, n$ columns. Then $\mathbf{u}_i$ and $\mathbf{v}_i$ are called the *left* and the *right* singular vectors of $A$ that correspond to the singular value $\sigma_i(A)$. Let

$$P_{A,L} := \sum_{i=1}^{\operatorname{rank} A} \mathbf{u}_i \mathbf{u}_i^* \in \mathbb{C}^{m\times m}, \quad P_{A,R} := \sum_{i=1}^{\operatorname{rank} A} \mathbf{v}_i \mathbf{v}_i^* \in \mathbb{C}^{n\times n} \tag{1.1}$$

be the orthogonal projections on the range of $A$ and $A^*$. Denote by

$$A_k := \sum_{i=1}^{k} \sigma_i(A) \mathbf{u}_i \mathbf{v}_i^* \in \mathbb{C}^{m\times n}$$

for $k = 1, \ldots, \operatorname{rank} A$. For $k > \operatorname{rank} A$, we define $A_k := A\ (= A_{\operatorname{rank} A})$. For $1 \leq k < \operatorname{rank} A$, the matrix $A_k$ is uniquely defined if and only if $\sigma_k(A) > \sigma_{k+1}(A)$.

The enormous application of the SVD in pure and applied mathematics is derived from the following approximation property:

$$\min_{X \in \mathcal{R}(m,n,k)} ||A - X||_F = ||A - A_k||_F, \quad k = 1, \ldots. \tag{1.2}$$

---

*Received by the editors March 28, 2006; accepted for publication (in revised form) by R. Mathias November 17, 2006; published electronically May 22, 2007.
http://www.siam.org/journals/simax/29-2/65551.html

[†] Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7045 (friedlan@uic.edu).

[‡] School of Mathematics and Statistics, University of South Australia, Mawson Lakes SA 5095, Australia (Anatoli.Torokhti@unisa.edu.au).

The latter is known as the Eckart–Young theorem [2]. We note that the work [2] implied a number of extensions and cite [4, 5, 7, 8] as some recent references. Another application of SVD is a formula for the Moore–Penrose inverse $A^\dagger := V_A \Sigma_A^\dagger U_A^* \in \mathbb{C}^{n\times m}$ of $A$, where

$$\Sigma_A^\dagger := \operatorname{diag}\left(\frac{1}{\sigma_1(A)}, \ldots, \frac{1}{\sigma_{\operatorname{rank} A}(A)}, 0, \ldots, 0\right) \in \mathbb{C}^{n\times m}.$$

See, for example, [1].

## 2. Main result.

Below, we provide generalizations of the classical minimal problem given in (1.2).

THEOREM 2.1. $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{m\times p}$ $C \in \mathbb{C}^{q\times n}$

$$X = B^\dagger (P_{B,L} A P_{C,R})_k C^\dagger \tag{2.1}$$

$$\min_{X\in\mathcal{R}(p,q,k)} ||A - BXC||_F, \tag{2.2}$$

$||X||_F$

$$k \geq \operatorname{rank} P_{B,L} A P_{C,R}$$

$$1 \leq k < \operatorname{rank} P_{B,L} A P_{C,R} \qquad \sigma_k(P_{B,L} A P_{C,R}) > \sigma_{k+1}(P_{B,L} A P_{C,R}).$$

2.1. Recall that the Frobenius norm is invariant under the multiplication from the left and right by the corresponding unitary matrices. Hence $||A - BXC||_F = ||\widetilde{A} - \Sigma_B \widetilde{X} \Sigma_C||_F$, where $\widetilde{A} := U_B^* A V_C$ and $\widetilde{X} := V_B^* X U_C$. Clearly, $X$ and $\widetilde{X}$ have the same rank and same Frobenius norm. Thus, it is enough to consider the minimal problem $\min_{\widetilde{X}\in\mathcal{R}(p,q,k)} ||\widetilde{A} - \Sigma_B \widetilde{X} \Sigma_C||_F$.

Let $s = \operatorname{rank} B$ and $t = \operatorname{rank} C$. Clearly, if $B$ or $C$ is a zero matrix, then $X = \mathbf{0}$ is the solution to the minimal problem (2.2). In this case, either $P_{B,L}$ or $P_{C,R}$ are zero matrices, and the theorem holds trivially in this case.

Let us consider the case $1 \leq s, 1 \leq t$. Define $B_1 := \operatorname{diag}(\sigma_1(B), \ldots, \sigma_s(B)) \in \mathbb{C}^{s\times s}, C_1 := \operatorname{diag}(\sigma_1(C), \ldots, \sigma_t(C)) \in \mathbb{C}^{t\times t}$. Partition $\widetilde{A}$ and $\widetilde{X}$ into four block matrices $A_{ij}$ and $X_{ij}$ with $i, j = 1, 2$ so that $\widetilde{A} = [A_{ij}]_{i,j=1}^2$ and $\widetilde{X} = [X_{ij}]_{i,j=1}^2$, where $A_{11}, X_{11} \in \mathbb{C}^{s\times t}$. (For certain values of $s$ and $t$, we may have to partition $\widetilde{A}$ or $\widetilde{X}$ to less than four block matrices.) Next, observe that $Z := \Sigma_B \tilde{X} \Sigma_C = [Z_{ij}]_{i,j=1}^2$, where $Z_{11} = B_1 X_{11} C_1$ and all other blocks $Z_{ij}$ are zero matrices. Since $B_1$ and $C_1$ are invertible we deduce that

$$\operatorname{rank} Z = \operatorname{rank} Z_{11} = \operatorname{rank} X_{11} \leq \operatorname{rank} \widetilde{X} \leq k.$$

The approximation property of $(A_{11})_k$ yields the inequality $||A_{11} - Z_{11}||_F \geq ||A_{11} - (A_{11})_k||_F$ for any $Z_{11}$ of, at most, rank $k$. Hence for any $Z$ of the above form,

$$||\widetilde{A} - Z||_F^2 = ||A_{11} - Z_{11}||_F^2 + \sum_{2<i+j\leq 4} ||A_{ij}||_F^2 \geq ||A_{11} - (A_{11})_k||_F^2 + \sum_{2<i+j\leq 4} ||A_{ij}||_F^2.$$

Thus, $\widehat{X} = [X_{ij}]_{i,j=1}^2$, where $X_{11} = B_1^{-1}(A_{11})_k C_1^{-1}$ and $X_{ij} = \mathbf{0}$ for all $(i,j) \neq (1,1)$ is a solution to the problem $\min_{\widetilde{X} \in \mathcal{R}(p,q,k)} ||\widetilde{A} - \Sigma_B \widetilde{X} \Sigma_C||_F$ with the minimal Frobenius norm. This solution is unique if and only if the solution $Z_{11} = (A_{11})_k$ is the unique solution to the problem $\min_{Z_{11} \in \mathcal{R}(s,t,k)} ||A_{11} - Z_{11}||_F$. This happens if either $k \geq \operatorname{rank} A_{11}$ or $1 \leq k < \operatorname{rank} A_{11}$ and $\sigma_k(A_{11}) > \sigma_{k+1}(A_{11})$. A straightforward calculation shows that $\widehat{X} = \Sigma_B^\dagger (P_{\Sigma_B,L} \widetilde{A} P_{\Sigma_C,R})_k \Sigma_C^\dagger$. Thus, a solution of (2.2) with the minimal Frobenius norm is given by

$$\begin{aligned} X &= B^\dagger U_B (P_{\Sigma_B,L} U_B^* A V_C P_{\Sigma_C,R})_k V_C^* C^\dagger \\ &= B^\dagger U_B (U_B^* P_{B,L} A P_{C,R} V_C)_k V_C^* C^\dagger \\ &= B^\dagger (P_{B,L} A P_{C,R})_k C^\dagger. \end{aligned}$$

This solution is unique if and only if either $k \geq \operatorname{rank} P_{B,L} A P_{C,R}$, or $1 \leq k < \operatorname{rank} P_{B,L} A P_{C,R}$ and $\sigma_k(P_{B,L} A P_{C,R}) > \sigma_{k+1}(P_{B,L} A P_{C,R})$. $\square$

**3. Examples.** First, observe that the classical approximation problem given by (1.2) is equivalent to the case $m = p, n = q, B = I_m, C = I_n$. (Here, $I_m$ is the $m \times m$ identity matrix.) Clearly, $P_{I_m,L} = I_m$, $P_{I_n,R} = I_n$, $I_m^\dagger = I_m$, $I_n^\dagger = I_n$. In this case we obtain the classical solution $B^\dagger (P_{B,L} A P_{C,R})_k C^\dagger = A_k$.

Second, if $p = m$, $q = n$, and $B$, $C$ are nonsingular, then $\operatorname{rank}(BXC) = \operatorname{rank} X$. In this case, $P_{B,L} = I_m$ and $P_{C,R} = I_n$, and the solution to (2.2) is given by $X = B^{-1} A_k C^{-1}$.

Next, a particular case of the problem (2.2) occurs in the study of a random vector estimation (see, for example, [6, 9]) as follows. Let $(\Omega, \Sigma, \mu)$ be a probability space, where $\Omega$ is the set of outcomes, $\Sigma$ a $\sigma$-field of measurable subsets of $\Omega$, and $\mu : \Sigma \mapsto [0,1]$ an associated probability measure on $\Sigma$ with $\mu(\Omega) = 1$. Suppose that $\mathbf{x} \in L^2(\Omega, \mathbb{R}^m)$ and $\mathbf{y} \in L^2(\Omega, \mathbb{R}^n)$ are random vectors such that $\mathbf{x} = (x_1, \ldots, x_m)^T$ and $\mathbf{y} = (y_1, \ldots, y_n)^T$ with $x_i, y_j \in L^2(\Omega, \mathbb{R})$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, respectively. Let $E_{xy} = [e_{ij,xy}] \in \mathbb{R}^{m \times n}$, $E_{yy} = [e_{jk,yy}] \in \mathbb{R}^{n \times n}$ be correlation matrices with entries

$$e_{ij,xy} = \int_\Omega x_i(\omega) y_j(\omega) d\mu(\omega), \quad e_{jk,yy} = \int_\Omega y_j(\omega) y_k(\omega) d\mu(\omega),$$
$$i = 1, \ldots, m, \quad j,k = 1, \ldots, n, \quad \omega \in \Omega.$$

The problems considered in [6, 9] are reduced to finding a solution to the problem (2.2) with $A = E_{xy} E_{yy}^{1/2\dagger}$, $B = I_n$, and $C = E_{yy}^{1/2}$, where we write $E_{yy}^{1/2\dagger} = (E_{yy}^{1/2})^\dagger$. Let the SVD of $E_{yy}^{1/2}$ be given by $E_{yy}^{1/2} = V_n \Sigma V_n^*$ and let $\operatorname{rank} E_{yy}^{1/2} = r$. Here, $V_n = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$ with $\mathbf{v}_i$ the $i$th column of $V_n$. By Theorem 2.1, the solution to this particular case of the problem (2.2) having the minimal Frobenius norm is given by $X = (E_{xy} E_{yy}^{1/2\dagger} V_r V_r^*)_k E_{yy}^{1/2\dagger}$, where $E_{yy}^{1/2\dagger} V_r V_r^* = E_{yy}^{1/2\dagger}$. Therefore, $X = (E_{xy} E_{yy}^{1/2\dagger})_k E_{yy}^{1/2\dagger}$. The conditions for the uniqueness follow directly from Theorem 2.1.

Finally, a special case of the minimal problem (2.2), where $X$ is a rank one matrix and $C$ the identity matrix, was considered by M. Elad [3] in the context of image processing.

## REFERENCES

[1] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications,* John Wiley & Sons, New York, 1974.

[2] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.

[3] M. Elad, *Personal communication*, 2005.

[4] A. Frieze, R. Kannan, and S. Vempala, *Fast Monte-Carlo algorithms for finding low-rank approximations*, J. ACM, 51 (2004), pp. 1025–1041.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computation*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[6] Y. Hua and W. Q. Liu, *Generalized Karhunen–Loève transform*, IEEE Signal Process. Lett., 6 (1998), pp. 141–142.

[7] T. G. Kolda, *A counterexample to the possibility of an extension of the Eckart–Young low-rank approximation theorem for the orthogonal rank tensor decomposition*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 762–767.

[8] W.-S. Lu, S.-C. Pei, and P.-H. Wang, *Weighted low-rank approximation of general complex matrices and its application in the design of 2-D digital filters*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 44 (1997), pp. 650–655.

[9] Y. Yamashita and H. Ogawa, *Relative Karhunen–Loève transform*, IEEE Trans. Signal Process., 44 (1996), pp. 371–378.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 660–671

# NORMS OF TOEPLITZ MATRICES WITH FISHER–HARTWIG SYMBOLS*

ALBRECHT BÖTTCHER† AND JANI VIRTANEN‡

**Abstract.** We describe the asymptotics of the spectral norm of finite Toeplitz matrices generated by functions with Fisher–Hartwig singularities as the matrix dimension goes to infinity. In the case of positive generating functions, our result provides the asymptotics of the largest eigenvalue, which is of interest in time series with long range memory.

**Key words.** Toeplitz matrix, spectral norm, Fisher–Hartwig singularity, time series, long range memory

**AMS subject classifications.** Primary, 47B35; Secondary, 15A60, 62M10, 62M20, 65F35

**DOI.** 10.1137/06066165X

## 1. Introduction.

Let $\{a_k\}_{k\in\mathbf{Z}}$ be a sequence of complex numbers and denote by $T_n$ the $n\times n$ Toeplitz matrix $(a_{j-k})_{j,k=0}^{n-1}$. We are interested in the behavior of the spectral norm $\|T_n\|$ as $n\to\infty$. Notice that if the matrix $T_n$ is positive definite, then $\|T_n\|$ is just the maximal eigenvalue of $T_n$.

If there is a function $a\in L^1(\mathbf{T})$ such that $\{a_k\}_{k\in\mathbf{Z}}$ is the sequence of the Fourier coefficients of $a$, that is, $a_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} a(e^{i\theta})e^{-ik\theta}d\theta$, we call $a$ the symbol of the sequence $\{T_n\}$ and denote $T_n$ by $T_n(a)$. The case where $a$ is in $L^\infty(\mathbf{T})$ is easy, since then $\|T_n(a)\|\to\|a\|_\infty$ as $n\to\infty$. Things are more complicated for symbols $a$ in $L^1(\mathbf{T})\setminus L^\infty(\mathbf{T})$. We here focus our attention on so-called Fisher–Hartwig symbols with a single singularity; that is, we consider functions $a$ of the form

$$a(t) = |t-t_0|^{-2\alpha}\varphi_{\beta,t_0}(t)b(t) \quad (t\in\mathbf{T}),$$

where $t_0\in\mathbf{T}$, $\alpha$ is a complex number subject to the constraint $0<\operatorname{Re}\alpha<1/2$, $\beta$ is a complex number satisfying $-1/2<\operatorname{Re}\beta\le 1/2$, the function $\varphi_{\beta,t_0}$ is defined as

$$\varphi_{\beta,t_0}(t) = \exp(i\beta\arg(-t/t_0)) \quad (t\in\mathbf{T})$$

with $\arg z\in(-\pi,\pi]$, and $b$ is a function in $L^\infty(\mathbf{T})$ that is continuous at $t_0$ and does not vanish at $t_0$. The hypothesis $0<\operatorname{Re}\alpha<1/2$ ensures that $a\in L^1(\mathbf{T})\setminus L^\infty(\mathbf{T})$. We should mention that if $a$ is any piecewise continuous function on $\mathbf{T}$ with a single jump, say, at $t_0\in\mathbf{T}$, and $a(t_0\pm 0)\ne 0$, then $a$ can be written in the form $a=\varphi_{\beta,t_0}b$ with $-1/2<\operatorname{Re}\beta\le 1/2$ and a continuous function $b$. Indeed, since $\varphi_{\beta,t_0}(t_0-0)=e^{\pi i\beta}$ and $\varphi_{\beta,t_0}(t_0+0)=e^{-\pi i\beta}$, it suffices to choose $\operatorname{Im}\beta\in(-\infty,\infty)$ and $\operatorname{Re}\beta\in(-1/2,1/2]$ so that

$$\frac{a(t_0+0)}{a(t_0-0)} = e^{-2\pi i\beta} = e^{2\pi\operatorname{Im}\beta}\,e^{-2\pi i\operatorname{Re}\beta}.$$

---

*Received by the editors June 1, 2006; accepted for publication (in revised form) by H. J. Woerdeman November 29, 2006; published electronically May 22, 2007.
http://www.siam.org/journals/simax/29-2/66165.html

†Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany (aboettch@mathematik.tu-chemnitz.de).

‡Department of Mathematics, University of Helsinki, Gustaf Hällströmin katu 2b, FI-00014 Helsinki, Finland (jani.virtanen@helsinki.fi). The research of this author was supported by the Academy of Finland Project 207048.

Our main result says that

$$\|T_n(a)\| \sim C_{\alpha,\beta}\, n^{2\,\mathrm{Re}\,\alpha}\, |b(t_0)| \quad \text{as} \quad n \to \infty,$$

where $C_{\alpha,\beta}$ is a completely identified constant depending only on $\alpha$ and $\beta$ and where $x_n \sim y_n$ means that $x_n/y_n \to 1$. We will also establish results for symbols with more than one Fisher–Hartwig singularity.

For the exciting story behind Toeplitz matrices with Fisher–Hartwig symbols and their determinants we refer the reader to the books [4], [5] and the papers [1], [9]. For general Toeplitz matrices, the asymptotic distribution of the singular values and the asymptotics of the extreme singular values have been studied by many authors, and we abstain from giving an ample list of references here. These investigations are mainly directed to the collective distribution of the singular values (Szegö–Avram–Parter theorems) or to the behavior of the extreme singular values of $T_n(a)$ for symbols $a$ in $L^\infty(\mathbf{T})$. The asymptotics of the smallest singular value are governed by the nature of the zeros of the symbol $a$. This implies that the rate of convergence of the largest singular value (= the norm) of $T_n(a)$ to $\|a\|_\infty$ depends on the zeros of the function $\|a\|_\infty - |a(t)|$. These results are not applicable to Toeplitz matrices with symbols in $L^1(\mathbf{T}) \setminus L^\infty(\mathbf{T})$ or to Toeplitz matrices "without symbols." Such matrices are considered in [2], [3], [12], [13], [14], [15], for example, but the focus of these papers is not on the problem we are interested in here.

Under the sole assumption that $b$ is in $L^\infty(\mathbf{T})$, the method of [2] yields the estimate $\|T_n(a)\| \le C_2\, n^{2\,\mathrm{Re}\,\alpha}$ with some finite constant $C_2$. If $\alpha$ is real, $\varphi_\beta b$ is real-valued, and $\operatorname{ess\,inf} b > 0$, one can also proceed as in [2] to show the existence of a positive constant $C_1$ such that $\|T_n(a)\| \ge C_1\, n^{2\,\mathrm{Re}\,\alpha}$. Such estimates were also derived in [11] by different arguments. These two-sided bounds are useful in several contexts (see [11], for example), but they are clearly far away from the precise asymptotics $\|T_n(a)\| \sim C_{\alpha,\beta}\, n^{2\,\mathrm{Re}\,\alpha}\, |b(t_0)|$.

The approach of the present paper is based on an idea of Widom [16], [17], [18]: we construct integral operators $K_n$ on $L^2(0,1)$ such that $\|T_n(a)\| = n^{2\,\mathrm{Re}\,\alpha}\, \|K_n\|$ and prove that $K_n$ converges to some integral operator $K$ in the operator norm on $L^2(0,1)$, which implies that $\|K_n\| \to \|K\|$.

For nonnegative symbols, the results of this paper are of interest in the analysis of time series with long memory. The $n$th covariance matrix of a time series is a positive definite Toeplitz matrix $T_n(a) = (a_{j-k})_{j,k=1}^n$, and one wants to know its largest eigenvalue. If the series has a short memory, then $a_n$ goes rapidly to zero as $|n| \to \infty$, and hence $\{a_n\}$ is the sequence of the Fourier coefficients of a function $a \in L^\infty(\mathbf{T})$. However, in the case of a long range memory, the numbers $a_n$ may be of the order $|n|^{2\alpha-1}$ $(0 < \alpha < 1/2)$, which leads to symbols $a \in L^1(\mathbf{T}) \setminus L^\infty(\mathbf{T})$. The symbol $a(t) = |t - t_0|^{-2\,\alpha}\, b(t)$ is especially popular and will be considered in detail in section 5. For more on Toeplitz matrices in time series we refer the reader to [6], [7], [8], [10], [11].

**2. A special class of Toeplitz matrices.** We begin with a simple observation.

PROPOSITION 2.1. [illegible] $\gamma$ [illegible] $|a_{\pm n}| = O(n^\gamma)$ [illegible] $n \to \infty$ [illegible] $\|T_n\|$ [illegible] $\gamma < -1$ $\|T_n\| = O(\log n)$ [illegible] $\gamma = -1$ [illegible] $\|T_n\| = O(n^{\gamma+1})$ [illegible] $\gamma > -1$ [illegible] $|a_{\pm n}| = o(n^\gamma)$ [illegible] $n \to \infty$ [illegible] $\|T_n\| = o(\log n)$ [illegible] $\gamma = -1$ [illegible] $\|T_n\| = o(n^{\gamma+1})$ [illegible] $\gamma > -1$

[illegible] In the case $\gamma < -1$, the sequence $\{a_k\}$ is the sequence of the Fourier coefficients of a continuous function $a$, and hence $\|T_n\| = \|T_n(a)\| \to \|a\|_\infty$. The spectral norm of a Toeplitz matrix one diagonal of which is occupied by units and the remaining diagonals of which are zero equals 1. This implies that $\|T_n\| \le \sum_{k=-(n-1)}^{n-1} |a_k|$ and therefore yields the assertions concerning $\gamma = -1$ and $\gamma > -1$. □

Let $A_n = (a_{j,k})_{j,k=0}^{n-1}$ be an $n \times n$ matrix with complex entries. We denote by $G_n$ the integral operator on $L^2(0,1)$ with the kernel

$$g_n(x,y) = a_{[nx],[ny]}, \quad (x,y) \in (0,1)^2,$$

where $[\xi]$ denotes the integral part of $\xi$.

Lemma 2.2 (Widom). [illegible] $A_n$ [illegible] $G_n$ [illegible] $\|A_n\| = n\|G_n\|$

[illegible] Put $I_k = (k/n, (k+1)/n)$ and consider the operators

$$S_n : \{x_k\}_{k=0}^{n-1} \mapsto \sqrt{n} \sum_{k=0}^{n-1} x_k \chi_{I_k}, \quad T_n : f \mapsto \left\{ \sqrt{n} \int_{I_k} f(x)dx \right\}_{k=0}^{n-1}.$$

It is easily seen that $\|S_n\| = \|T_n\| = 1$ and that $T_n S_n$ is the identity operator on $\mathbf{C}^n$. Since $S_n A_n T_n = nG_n$ and thus $A_n = nT_n G_n S_n$ we obtain that $\|A_n\| \ge n\|G_n\|$ and $\|A_n\| \le n\|G_n\|$. □

Let $C^+$, $C^-$, $\gamma$ be complex numbers, let $\operatorname{Re}\gamma > -1$, let $a_{\pm n} = C^{\pm} n^{\gamma}$ for $n \ge 1$, and let $a_0$ be any complex number. Denote by $K_n$ and $K$ the integral operators on $L^2(0,1)$ with the kernels

$$k_n(x,y) = n^{-\gamma}\, a_{[nx]-[ny]} \quad \text{and} \quad k(x,y) = \begin{cases} C^+ (x-y)^{\gamma} & \text{for} \quad x > y, \\ C^-(y-x)^{\gamma} & \text{for} \quad x < y, \end{cases}$$

respectively.

Lemma 2.3. [illegible] $K_n$ [illegible] $K$ [illegible] $L^2(0,1)$

[illegible] Fix a $\mu \in (0,1)$ sufficiently close to 1 such that $(1-\mu)|\operatorname{Re}\gamma| < \mu$ and $2\,\mu \operatorname{Re}\gamma < 1 + 2\operatorname{Re}\gamma$. Put

$$k_n^1(x,y) = \begin{cases} k(x,y) & \text{if } |x-y| > n^{\mu-1}, \\ 0 & \text{otherwise}, \end{cases} \quad k_n^2(x,y) = \begin{cases} k(x,y) & \text{if } |x-y| < n^{\mu-1}, \\ 0 & \text{otherwise}, \end{cases}$$

$$\ell_n^1(x,y) = \begin{cases} k_n(x,y) & \text{if } |x-y| > n^{\mu-1}, \\ 0 & \text{otherwise}, \end{cases} \quad \ell_n^2(x,y) = \begin{cases} k_n(x,y) & \text{if } |x-y| < n^{\mu-1}, \\ 0 & \text{otherwise}, \end{cases}$$

and denote by $K_n^1, K_n^2, L_n^1, L_n^2$ the integral operators on $L^2(0,1)$ with the kernels $k_n^1, k_n^2, \ell_n^1, \ell_n^2$, respectively. We have $K = K_n^1 + K_n^2$ and $K_n = L_n^1 + L_n^2$. Thus,

$$\|K - K_n\| \le \|K_n^1 - L_n^1\| + \|K_n^2\| + \|L_n^2\|.$$

We show that each term on the right goes to zero as $n \to \infty$.

To prove that $\|K^1 - K_n^1\| \to 0$ it suffices to show that $|k_n^1(x,y) - \ell_n^1(x,y)|$ converges uniformly to zero for $|x-y| > n^{\mu-1}$. We may assume that $x > y$, since the case $x < y$ can be tackled analogously. Thus, let $x - y > n^{\mu-1}$. As $[nx] - [ny] = n(x-y) + \varepsilon_n$ with $|\varepsilon_n| = |\varepsilon_n(x,y)| \le 2$, we get

$$\begin{aligned} \ell_n^1(x,y) &= C^+ n^{-\gamma}([nx] - [ny])^{\gamma} = C^+ n^{-\gamma}(n(x-y) + \varepsilon_n)^{\gamma} \\ &= C^+ (x-y)^{\gamma} \left(1 + \frac{\varepsilon_n}{n(x-y)}\right)^{\gamma}. \end{aligned}$$

Since $n(x-y) > n^\mu$, it follows that $\ell_n^1(x,y) = C^+(x-y)^\gamma(1+O(n^{-\mu}))$ uniformly in $x$ and $y$. Hence

$$|k_n^1(x,y) - \ell_n^1(x,y)| = |(x-y)^\gamma|\, O(n^{-\mu}) = (x-y)^{\operatorname{Re}\gamma}\, O(n^{-\mu})$$

uniformly in $x$ and $y$. If $\operatorname{Re}\gamma \ge 0$, this goes to zero uniformly in $x$ and $y$. In the case where $\operatorname{Re}\gamma < 0$, we use the inequality $x - y > n^{\mu-1}$ to obtain that

$$(x-y)^{\operatorname{Re}\gamma}\, O(n^{-\mu}) = O\left(n^{(1-\mu)|\operatorname{Re}\gamma|} n^{-\mu}\right)$$

uniformly in $x$ and $y$, which is $o(1)$ because $(1-\mu)|\operatorname{Re}\gamma| < \mu$. We so have proved that $\|K^1 - K_n^1\| \to 0$ as $n \to \infty$.

The operator $K_n^2$ is the compression to $L^2(0,1)$ of the operator of convolution on $L^2(\mathbf{R})$ by the kernel

$$\kappa(x) = \begin{cases} C^+ x^\gamma & \text{for } \; 0 < x < n^{\mu-1}, \\ C^-|x|^\gamma & \text{for } \; -n^{\mu-1} < x < 0, \\ 0 & \text{for } \; |x| > n^{\mu-1}. \end{cases}$$

The norm of a convolution operator on $L^2(\mathbf{R})$ is the maximum of the modulus of the Fourier transform

$$(F\kappa)(\xi) = \int_{\mathbf{R}} \kappa(x) e^{i\xi x} dx \quad (\xi \in \mathbf{R})$$

of its convolution kernel $\kappa(x)$. Hence

$$\begin{aligned} \|K_n^2\| &\le \max_{\xi\in\mathbf{R}} |(F\kappa)(\xi)| \le \int_{\mathbf{R}} |\kappa(x)| dx \\ &= \int_{-n^{\mu-1}}^{1} C^-|x|^{\operatorname{Re}\gamma} dx + \int_0^{n^{\mu-1}} C^+ x^{\operatorname{Re}\gamma} dx = O\left(n^{(\mu-1)(\operatorname{Re}\gamma+1)}\right), \end{aligned}$$

which proves that $\|K_n^2\| \to 0$ as $n \to \infty$.

Let us consider the norm $\|L_n^2\|$. The kernel $\ell_n^2(x,y)$ is supported in the strip $|x-y| < n^{\mu-1}$. Let $\tilde{\ell}_n^2(x,y)$ be $k_n(x,y)$ for $(x,y)$ in the staircase-like bordered strip $|[nx] - [ny]| < n^\mu$ and zero otherwise. Denote by $\tilde{L}_n^2$ the corresponding integral operator. The difference $\ell_n^2(x,y) - \tilde{\ell}_n^2(x,y)$ is supported in about $4(n - n^\mu) = O(n)$ squares of side length $1/n$, and in these squares the absolute value of the difference is about $n^{-\gamma} a_{\pm[n^\mu]} = O(n^{-\operatorname{Re}\gamma} n^{\mu \operatorname{Re}\gamma})$. Consequently, the squared Hilbert–Schmidt norm $\|L_n^2 - \tilde{L}_n^2\|_2^2$ is at most a constant times $n\, n^{-2\operatorname{Re}\gamma} n^{2\mu\operatorname{Re}\gamma}(1/n)^2$, which goes to zero because $1 - 2\operatorname{Re}\gamma + 2\mu\operatorname{Re}\gamma - 2 = 2\mu\operatorname{Re}\gamma - (1 + 2\operatorname{Re}\gamma) < 0$. We are therefore left with proving that $\|\tilde{L}_n^2\| \to 0$. Let $T_n = (b_{j-k})_{j,k=0}^{n-1}$, where $b_k = a_k$ for $|k| \le n^\mu$ and $b_k = 0$ otherwise. Lemma 2.2 implies that $\|\tilde{L}_n^2\| = (1/n) n^{-\operatorname{Re}\gamma}\|T_n\|$, and since

$$\|T_n\| \le \sum_{k=-n^\mu}^{n^\mu} |b_k| = O\left(\sum_{k=-n^\mu}^{n^\mu} k^{\operatorname{Re}\gamma}\right) = O\left(n^{\mu(\operatorname{Re}\gamma+1)}\right),$$

we finally get $\|\tilde{L}_n^2\| = O\left(n^{(\mu-1)(\operatorname{Re}\gamma+1)}\right) = o(1)$. □

THEOREM 2.4. [illegible] $T_n = (a_{j-k})_{j,k=0}^{n-1}$ [illegible] $a_{\pm n} = C^{\pm} n^\gamma (1 + o(1))$ [illegible] $n \to \infty$ [illegible] $C^+$ $C^-$ $\gamma$ [illegible] $\operatorname{Re}\gamma > -1$ [illegible] $C^+$ [illegible] $C^-$ [illegible]

$$\|T_n\| \sim \|K\|\, n^{\operatorname{Re}\gamma+1},$$

[illegible] $K$ [illegible] $L^2(0,1)$ [illegible] $C^+(x-y)^\gamma$ [illegible] $x>y$ [illegible] $C^-(y-x)^\gamma$ [illegible] $x<y$

[illegible] Write $T_n = S_n + D_n$ with $S_n = (b_{j-k})_{j,k=0}^{n-1}$, $D_n = (d_{j-k})_{j,k=0}^{n-1}$, $b_{\pm n} = C^\pm n^\gamma$, $d_{\pm n} = o(n^\gamma)$. From Lemma 2.2 we deduce that $\|S_n\|/n$ equals the norm of the integral operator $n^\gamma K_n$, where $K_n$ has the kernel $n^{-\gamma} b_{[nx]-[ny]}$. Lemma 2.3 implies that $\|K_n - K\| \to 0$ and thus $\|K_n\| \to \|K\|$. Consequently,

$$\|S_n\| = \|K\| n^{\operatorname{Re}\gamma+1}(1+o(1)).$$

Proposition 2.1 yields $\|D_n\| = o(n^{\operatorname{Re}\gamma+1})$. □

THEOREM 2.5. [illegible] $B_1^+, \dots, B_Q^+, B_1^-, \dots, B_Q^-$ [illegible] $\gamma_1, \dots, \gamma_Q$ [illegible] $\operatorname{Re}\gamma_s > -1$ [illegible] $s$ [illegible] $\omega = e^{2\pi i/Q}$ [illegible] $T_n = (a_{j-k})_{j,k=0}^{n-1}$ [illegible]

$$a_{\pm n} = \sum_{s=1}^{Q} B_s^\pm \omega^{\pm sn} n^{\gamma_s}(1+o(1)) \quad n \to \infty.$$

[illegible] $\operatorname{Re}\gamma := \max_s \operatorname{Re}\gamma_s$ [illegible] $S = \{s : \operatorname{Re}\gamma_s = \operatorname{Re}\gamma\}$ [illegible]

$$\|T_n\| \sim \max_{s\in S} \|K_s\| \, n^{\operatorname{Re}\gamma+1},$$

[illegible] $K_s$ [illegible] $L^2(0,1)$ [illegible] $B_s^+(x-y)^{\gamma_s}$ [illegible] $x>y$ [illegible] $B_s^-(y-x)^{\gamma_s}$ [illegible] $x<y$

[illegible] Assume first that $n = mQ$ with a natural number $m$. We rearrange the rows of $T_{mQ}$ by first taking the rows $1, Q+1, 2Q+1, \dots$, then the rows $2, Q+2, 2Q+2, \dots$, and so on. Then we make the same rearrangement with the columns. The resulting matrix has the same spectral norm as $T_{mQ}$ and is a block Toeplitz matrix $(A_{j-k})_{j,k=0}^{Q-1}$ whose blocks are the Toeplitz matrices $A_k = (a_{k+(u-v)Q})_{u,v=0}^{m-1}$. For $0 \le |k| \le Q-1$, let $D_k$ be the Toeplitz matrix $D_k = (d_{u-v}^{(k)})_{u,v=0}^{m-1}$ given by $d_0^{(k)} = 0$ and

$$d_{\pm\nu}^{(k)} = \sum_{s=1}^{Q} B_s^\pm \omega^{sk} (\nu Q)^{\gamma_s} \quad (\nu \ge 1).$$

If $\nu \to \infty$, then eventually $k + \nu Q \ge 1$, and hence

$$a_{k+\nu Q} - d_\nu^{(k)} = \sum_{s=1}^{Q} B_s^+ \omega^{sk} \left[ (k+\nu Q)^{\gamma_s}(1+o(1)) - (\nu Q)^{\gamma_s} \right].$$

The modulus of the term in brackets is

$$\begin{aligned} &(\nu Q)^{\operatorname{Re}\gamma_s} \left| \left(1 + \frac{k}{\nu Q}\right)^{\gamma_s} (1+o(1)) - 1 \right| \\ &= (\nu Q)^{\operatorname{Re}\gamma_s} |(1+o(1))(1+o(1)) - 1| = (\nu Q)^{\operatorname{Re}\gamma_s} o(1) = o(\nu^{\operatorname{Re}\gamma_s}). \end{aligned}$$

An analogous estimate holds for $\nu \to -\infty$. From Proposition 2.1 we therefore deduce that $A_k = D_k + E_k$ with $\|E_k\| = o(m^{\operatorname{Re}\gamma+1})$. It follows that

$$\|T_{mQ}\| = \|(D_{j-k})_{j,k=0}^{Q-1}\| + o(m^{\operatorname{Re}\gamma+1}).$$

Now, for $s \in \{1, \ldots, Q\}$, put $H_s = (h^{(s)}_{u-v})_{u,v=0}^{m-1}$, where $h^{(s)}_0 = 0$ and $h^{(s)}_{\pm\nu} = B^{\pm}_s(\nu Q)^{\gamma_s}$ for $\nu \geq 1$. Then

$$D_{j-k} = \sum_{s=1}^{Q} \omega^{s(j-k)} H_s.$$

Let $F_\pm$ be the block Fourier matrices $F_\pm = (\omega^{\pm jk} I_{m\times m})_{j,k=1}^{Q}$. The $j, k$ block entry of $F_+ \operatorname{diag}(H_1, \ldots, H_Q) F_-$ equals $\sum_{s=1}^{Q} \omega^{js} H_s \omega^{-sk} = D_{j-k}$. Consequently,

$$(D_{j-k})_{j,k=0}^{Q-1} = (D_{j-k})_{j,k=1}^{Q} = F_+ \operatorname{diag}(H_1, \ldots, H_Q) F_-.$$

Since the matrices $(1/\sqrt{Q})\, F_\pm$ are unitary, we get

$$\|(D_{j-k})_{j,k=0}^{Q-1}\| = \sqrt{Q}\ \|\operatorname{diag}(H_1, \ldots, H_Q)\|\ \sqrt{Q} = Q \max_s \|H_s\|.$$

Theorem 2.4 gives

$$\|H_s\| = m^{\operatorname{Re}\gamma_s}\, Q^{\operatorname{Re}\gamma_s + 1}\, \|K_s\|\, (1 + o(1)).$$

In summary,

$$\begin{aligned} \|T_{mQ}\| &= Q \max_s m^{\operatorname{Re}\gamma_s+1}\, Q^{\operatorname{Re}\gamma_s}\, \|K_s\|\, (1+o(1)) + o(m^{\operatorname{Re}\gamma+1}) \\ &= (mQ)^{\operatorname{Re}\gamma+1} \max_{s\in S} \|K_s\| + o(m^{\operatorname{Re}\gamma+1}) \\ &\sim (mQ)^{\operatorname{Re}\gamma+1} \max_{s\in S} \|K_s\|. \end{aligned}$$

Finally, if $n$ is not divisible by $Q$, we can obtain $T_n$ from $T_{mQ}$ by adding at most $Q-1$ rows and columns. The spectral norm of a matrix with a single nonzero row or column is the $\ell^2$ norm of this row or column, which in the case at hand does not exceed the square root of $\sum_{j=-(n-1)}^{n-1} |a_j|^2 = O(n^{2\operatorname{Re}\gamma+1})$; that is, $O(n^{\operatorname{Re}\gamma+1/2}) = o(n^{\operatorname{Re}\gamma+1})$. This completes the proof. $\square$

**3. A single Fisher–Hartwig singularity.** We first consider the pure Fisher–Hartwig singularity at $t_0 = 1$, that is, the function

$$\sigma(t) = |t-1|^{-2\alpha} \varphi_{\beta,1}(t)$$

with $0 < \operatorname{Re}\alpha < 1/2$ and $-1/2 < \operatorname{Re}\beta \leq 1/2$. The Fourier coefficients of $\sigma$ are

$$\sigma_n = (-1)^n \frac{\Gamma(1-2\alpha)}{\Gamma(-\alpha+\beta+1-n)\Gamma(-\alpha-\beta+1+n)},$$

with the convention that $\sigma_n := 0$ for $n < 0$ if $\alpha = -\beta$ and $\sigma_n := 0$ for $n > 0$ if $\alpha = \beta$ (see [5, Lemma 6.18]). Using the formula

$$\Gamma(1-z) = \frac{\pi z}{\sin \pi z} \frac{1}{\Gamma(1+z)}$$

we see that

$$\begin{aligned} \sigma_n &= (-1)^n\, \Gamma(1-2\alpha) \frac{\sin \pi(n+\alpha-\beta)}{\pi(n+\alpha-\beta)} \frac{\Gamma(n+1+\alpha-\beta)}{\Gamma(n+1-\alpha-\beta)} \\ &= \Gamma(1-2\alpha) \frac{\sin \pi(\alpha-\beta)}{\pi(n+\alpha-\beta)} \frac{\Gamma(n+1+\alpha-\beta)}{\Gamma(n+1-\alpha-\beta)} \end{aligned}$$

for $n \geq 0$ and

$$\begin{aligned}\sigma_{-n} &= (-1)^n \frac{\Gamma(1-2\alpha)}{\Gamma(-\alpha+\beta+1+n)\Gamma(-\alpha-\beta+1-n)} \\ &= (-1)^n \,\Gamma(1-2\alpha)\,\frac{\sin\pi(n+\alpha+\beta)}{\pi(n+\alpha+\beta)}\,\frac{\Gamma(n+1+\alpha+\beta)}{\Gamma(n+1-\alpha+\beta)} \\ &= \Gamma(1-2\alpha)\,\frac{\sin\pi(\alpha+\beta)}{\pi(n+\alpha+\beta)}\,\frac{\Gamma(n+1+\alpha+\beta)}{\Gamma(n+1-\alpha+\beta)}\end{aligned}$$

for $n \geq 0$. The asymptotic formula $\Gamma(n+\gamma)/\Gamma(n+\delta) \sim n^{\gamma-\delta}$ $(n\to\infty)$ shows that

$$\sigma_n = C^{+}_{\alpha,\beta}\, n^{2\alpha-1}(1+o(1)), \quad \sigma_{-n} = C^{-}_{\alpha,\beta}\, n^{2\alpha-1}(1+o(1))$$

as $n \to \infty$, where

$$C^{\pm}_{\alpha,\beta} = \Gamma(1-2\alpha)\,\frac{\sin\pi(\alpha\mp\beta)}{\pi}.$$

We denote by $K$ the integral operator on $L^2(0,1)$ with the kernel

$$k(x,y) = \begin{cases} C^{+}_{\alpha,\beta}\,(x-y)^{2\alpha-1} & \text{for} \quad x>y, \\ C^{-}_{\alpha,\beta}\,(y-x)^{2\alpha-1} & \text{for} \quad x<y. \end{cases}$$

Obviously, $\|K\| > 0$.

THEOREM 3.1. [illegible] $\sigma(t) = |t-t_0|^{-2\alpha}\varphi_{\beta,t_0}(t)$ [illegible] $t_0 \in \mathbf{T}$ $0 < \operatorname{Re}\alpha < 1/2$ $-1/2 < \operatorname{Re}\beta \leq 1/2$ [illegible]

$$\|T_n(\sigma)\| \sim \|K\|\, n^{2\operatorname{Re}\alpha}.$$

[illegible] We change notation slightly and denote the function $\sigma$ defined as $\sigma(t) = |t-1|^{-2\alpha}\varphi_{\beta,1}(t)$ by $\sigma^0$. The $\sigma$ of the present theorem results from $\sigma^0$ by replacing $t_0 = 1$ with a general $t_0 \in \mathbf{T}$. The only change in the Fourier coefficients is that the $(-1)^n$ in $(\sigma^0)_n$ becomes $(-1/t_0)^n$ in $\sigma_n$, and hence $T_n(\sigma) = \Lambda\, T_n(\sigma^0)\,\Lambda^{-1}$, where $\Lambda := \operatorname{diag}(1, t_0^{-1}, \ldots, t_0^{-(n-1)})$. Therefore $\|T_n(\sigma)\| = \|T_n(\sigma^0)\|$. Taking into account that $(\sigma^0)_{\pm n} = C^{\pm}_{\alpha,\beta}\, n^{2\,\alpha-1}\,(1+o(1))$ and using Theorem 2.4 we arrive at the desired formula. □

PROPOSITION 3.2. [illegible] $\sigma$ [illegible] 3.1 [illegible] $c \in L^\infty(\mathbf{T})$ [illegible] $t_0$ [illegible]

$$\|T_n(\sigma c)\| = o(n^{2\operatorname{Re}\alpha}).$$

[illegible] Without loss of generality assume that $t_0 = 1$. Writing $\sigma = \operatorname{Re}\sigma + i\operatorname{Im}\sigma$ and $c = \operatorname{Re} c + i \operatorname{Im} c$ we get $T_n(\sigma c) = T_n(\operatorname{Re}\sigma \operatorname{Re} c) + \ldots$ (four terms) and thus $\|T_n(\sigma c)\| \leq \|T_n(\operatorname{Re}\sigma\operatorname{Re} c)\| + \ldots$. The matrix $T_n(\operatorname{Re}\sigma\operatorname{Re} c)$ is Hermitian, and hence

$$\begin{aligned}\|T_n(\operatorname{Re}\sigma\operatorname{Re} c)\| &= \max_{\psi\in\mathbf{C}^n\setminus\{0\}} \frac{|(T_n(\operatorname{Re}\sigma\operatorname{Re} c)\psi,\psi)|}{\|\psi\|^2} \\ &= \max_{\varphi\in\mathcal{P}_n\setminus\{0\}} \frac{1}{\|\varphi\|^2}\left|\int_{-\pi}^{\pi} \operatorname{Re}\sigma(x)\operatorname{Re} c(x)\,|\varphi(x)|^2\frac{dx}{2\pi}\right| \\ &\leq \max_{\varphi\in\mathcal{P}_n\setminus\{0\}} \frac{1}{\|\varphi\|^2}\int_{-\pi}^{\pi} |\operatorname{Re}\sigma(x)|\,|\operatorname{Re} c(x)|\,|\varphi(x)|^2\frac{dx}{2\pi},\end{aligned}$$

where $\mathcal{P}_n$ is the set of all trigonometric polynomials of the form $\varphi(x) = \varphi_0 + \varphi_1 e^{ix} + \cdots + \varphi_{n-1} e^{i(n-1)x}$. Notice that

$$\begin{aligned}\|\varphi\|_\infty^2 &= \|\varphi_0 + \varphi_1 e^{ix} + \cdots + \varphi_{n-1} e^{i(n-1)x}\|_\infty^2 \le (|\varphi_0| + |\varphi_1| + \cdots + |\varphi_{n-1}|)^2 \\ &\le n\,(|\varphi_0|^2 + |\varphi_1|^2 + \cdots + |\varphi_{n-1}|^2) = \frac{n}{2\pi}\int_{-\pi}^{\pi} |\varphi(x)|^2 dx = \frac{n}{2\pi}\,\|\varphi\|^2.\end{aligned}$$

Clearly,

$$\operatorname{Re}\sigma(x) = \left|2\,\sin\frac{x}{2}\right|^{-2\operatorname{Re}\alpha} \cos\left(\operatorname{Im}\alpha\,\log\left|2\sin\frac{x}{2}\right|\right)\,|\varphi_{\beta,1}(x)|,$$

which is $O(|x|^{-2\operatorname{Re}\alpha})$ as $x \to 0$. We split the integral into $\int_{|x|<\pi/n}$, $\int_{\pi/n<|x|<\pi/\sqrt{n}}$, and $\int_{\pi/\sqrt{n}<|x|<\pi}$. The integral over $|x| < \pi/n$ is at most

$$\begin{aligned}&C_1 \sup_{|x|<\pi/n} |\operatorname{Re} c(x)|\,\|\varphi\|_\infty^2 \int_{|x|<\pi/n} |x|^{-2\operatorname{Re}\alpha} dx \\ &\le C_2 \sup_{|x|<\pi/n} |\operatorname{Re} c(x)|\,\frac{n}{2\pi}\,\|\varphi\|^2\, n^{2\operatorname{Re}\alpha - 1} = o(n^{2\operatorname{Re}\alpha})\,\|\varphi\|^2\end{aligned}$$

because $\operatorname{Re} c(x) \to 0$ as $x \to 0$; here sup means esssup. The integral over the interval $\pi/n < |x| < \pi/\sqrt{n}$ has the upper bound

$$\begin{aligned}&C_3 \sup_{|x|<\pi/\sqrt{n}} |\operatorname{Re} c(x)| \int_{|x|>\pi/n} |x|^{-2\operatorname{Re}\alpha} |\varphi(x)|^2 dx \\ &\le C_3 \sup_{|x|<\pi/\sqrt{n}} |\operatorname{Re} c(x)|\,\frac{n^{2\operatorname{Re}\alpha}}{\pi^{2\operatorname{Re}\alpha}} \int_{|x|>\pi/n} |\varphi(x)|^2 dx \\ &\le C_3 \sup_{|x|<\pi/\sqrt{n}} |\operatorname{Re} c(x)|\,\frac{n^{2\operatorname{Re}\alpha}}{\pi^{2\operatorname{Re}\alpha}}\,\|\varphi\|^2 = o(n^{2\operatorname{Re}\alpha})\,\|\varphi\|^2,\end{aligned}$$

again because $\operatorname{Re} c(x) \to 0$ as $x \to 0$. Finally, the integral over $|x| > \pi/\sqrt{n}$ does not exceed

$$\begin{aligned}&C_4\,\|\operatorname{Re} c\|_\infty \int_{|x|>\pi/\sqrt{n}} |x|^{-2\operatorname{Re}\alpha} |\varphi(x)|^2 dx \\ &\le C_4\,\|\operatorname{Re} c\|_\infty\,\frac{n^{\operatorname{Re}\alpha}}{\pi^{2\operatorname{Re}\alpha}} \int_{|x|>\pi/\sqrt{n}} |\varphi(x)|^2 dx \\ &\le C_4\,\|\operatorname{Re} c\|_\infty\,\frac{n^{\operatorname{Re}\alpha}}{\pi^{2\operatorname{Re}\alpha}}\,\|\varphi\|^2 = O(n^{\operatorname{Re}\alpha})\,\|\varphi\|^2 = o(n^{2\operatorname{Re}\alpha})\,\|\varphi\|^2.\end{aligned}$$

This proves that $\|T_n(\operatorname{Re}\sigma\operatorname{Re} c)\| = o(n^{2\operatorname{Re}\alpha})$. Analogously one can show that

$$\|T_n(\operatorname{Re}\sigma\operatorname{Im} c)\|, \quad \|T_n(\operatorname{Im}\sigma\operatorname{Re} c)\|, \quad \|T_n(\operatorname{Im}\sigma\operatorname{Im} c)\|$$

are $o(n^{2\operatorname{Re}\alpha})$. □

THEOREM 3.3. [illegible] $a = \sigma b$ [illegible] $\sigma$ [illegible] 3.1 [illegible] $b$ [illegible] $L^\infty(\mathbf{T})$ [illegible] $t_0$ [illegible] $t_0$ [illegible]

$$\|T_n(a)\| \sim \|K\|\,|b(t_0)|\,n^{2\operatorname{Re}\alpha}.$$

[illegible] We have $b(t) = b(t_0) + c(t)$ with $b(t_0) \neq 0$ and a function $c \in L^\infty(\mathbf{T})$ that is continuous and zero at $t_0$. It follows that $T_n(a) = b(t_0)\, T_n(\sigma) + T_n(\sigma c)$. Theorem 3.1 yields

$$\|b(t_0)\, T_n(\sigma)\| = |b(t_0)|\, \|T_n(\sigma)\| = |b(t_0)|\, \|K\|\, n^{2\,\mathrm{Re}\,\alpha}\,(1 + o(1)),$$

and Proposition 3.2 gives $\|T_n(\sigma c)\| = o(n^{2\,\mathrm{Re}\,\alpha})$. □

**4. Several Fisher–Hartwig singularities.** Let $R \geq 2$ and

$$a(t) = b(t) \prod_{r=1}^{R} |t - t_r|^{-2\alpha_r} \varphi_{\beta_r,t_r}(t) \quad (t \in \mathbf{T}),$$

where $t_1, \ldots, t_R$ are distinct points on $\mathbf{T}$, $0 < \mathrm{Re}\,\alpha_r < 1/2$, $-1/2 < \mathrm{Re}\,\beta_r \leq 1/2$, $b \in L^\infty(\mathbf{T})$, $b$ is continuous at the points $t_1, \ldots, t_R$, and $b(t_r) \neq 0$ for all $r$. It is easily seen that $a$ can be written in the form

$$a(t) = \sum_{r=1}^{R} |t - t_r|^{-2\alpha_r} \varphi_{\beta_r,t_r}(t)\, b_r(t) \quad (t \in \mathbf{T})$$

with functions $b_r \in L^\infty(\mathbf{T})$ such that $b_r$ is continuous at $t_r$ and satisfies $b_r(t_r) \neq 0$. Let

$$\mathrm{Re}\,\alpha := \max\{\mathrm{Re}\,\alpha_1, \ldots, \mathrm{Re}\,\alpha_R\}, \qquad M = \{r : \mathrm{Re}\,\alpha_r = \mathrm{Re}\,\alpha\}.$$

If there is only one $r_0$ such that $\mathrm{Re}\,\alpha_{r_0} = \mathrm{Re}\,\alpha$, then Theorem 3.3 implies that

$$\|T_n(a)\| \sim \|K_{r_0}\|\, |b_{r_0}(t_{r_0})|\, n^{2\,\mathrm{Re}\,\alpha},$$

where $K_r$ denotes the integral operator on $L^2(0,1)$ associated with $|t - t_r|^{-2\alpha_r} \varphi_{\beta_r,t_r}(t)$, that is, the integral operator whose kernel is $C^+_{\alpha_r,\beta_r}(x-y)^{2\,\alpha_r - 1}$ for $x > y$ and equals $C^-_{\alpha_r,\beta_r}(y-x)^{2\,\alpha_r - 1}$ for $x < y$. The case where the maximum is attained at more than one $r$ is more difficult.

CONJECTURE 4.1. [illegible]

$$\|T_n(a)\| \sim \max_{r \in M} \|K_r\|\, |b(t_r)|\, n^{2\,\mathrm{Re}\,\alpha}.$$

The following result confirms this conjecture in a sufficiently interesting special case.

THEOREM 4.2. [illegible] $t_0 \in \mathbf{T}$ [illegible] $r$ $t_r = e^{2\pi i \varphi_r} t_0$ [illegible] $\varphi_r$ [illegible] 4.1 [illegible]

[illegible] As passage from $a(t)$ to $a(t/t_0)$ does not change the spectral norm of the Toeplitz matrix (recall the proof of Theorem 3.1), we may without loss of generality assume that $t_0 = 1$. Put $\sigma_{\alpha,\beta,\tau}(t) = |t - \tau|^{-2\alpha} \varphi_{\beta,\tau}(t)$. The Fourier coefficients of $a$ are

$$a_n = \sum_{r=1}^{R} (\sigma_{\alpha_r,\beta_r,t_r}\, b_r)_n = \sum_{r=1}^{R} b_r(t_r)\, (\sigma_{\alpha_r,\beta_r,t_r})_n + f_n,$$

where $\{f_n\}$ is the sequence of the Fourier coefficients of a function $f \in L^1(\mathbf{T})$ for which $\|T_n(f)\| = o(n^{2\,\mathrm{Re}\,\alpha})$ (Proposition 3.2). Furthermore, $(\sigma_{\alpha_r,\beta_r,t_r})_n = t_r^{-n} (\sigma_{\alpha_r,\beta_r,1})_n$ (see once more the proof of Theorem 3.1). Thus,

$$a_n = \sum_{r=1}^{R} b_r(t_r)\, t_r^{-n}\, (\sigma_{\alpha_r,\beta_r,1})_n + f_n.$$

Let $t_r^{-1} = e^{2\pi i p_r/q_r}$ with a rational number $p_r/q_r \in (0,1]$ and denote by $Q$ the least common multiple of $q_1, \dots, q_R$. Put $\omega = e^{2\pi i/Q}$. Then each $t_r^{-1}$ is of the form $\omega^{k_r}$ with some $k_r \in \{1, 2, \dots, Q\}$. It follows that

$$a_n = \sum_{r=1}^{R} b_r(t_r)\, \omega^{k_r n}\, (\sigma_{\alpha_r,\beta_r,1})_n + f_n$$

with different $k_1, \dots, k_R$ belonging to $\{1, 2, \dots, Q\}$. From section 3 we know that

$$(\sigma_{\alpha,\beta,1})_{\pm n} = C^{\pm}_{\alpha,\beta}\, n^{2\alpha-1}(1 + o(1)).$$

Hence

$$a_{\pm n} = \sum_{r=1}^{R} b_r(t_r)\, \omega^{\pm k_r n}\, C^{\pm}_{\alpha_r,\beta_r}\, n^{2\alpha_r-1}(1 + o(1)) + f_n,$$

which can be written as

$$a_{\pm n} = \sum_{s=1}^{Q} B^{\pm}_s\, \omega^{\pm sn}\, n^{\gamma_s}(1 + o(1)) + f_n$$

with $B^{\pm}_{k_r} = b_r(t_r)\, C^{\pm}_{\alpha_r,\beta_r}$, $\gamma_{k_r} = 2\alpha_r - 1$ and $B^{\pm}_s = 0$, $\gamma_s = 0$ otherwise. Theorem 2.5 shows that the spectral norm of the Toeplitz matrix $T^0_n$ generated by

$$a^0_{\pm n} := \sum_{s=1}^{Q} B^{\pm}_s\, \omega^{\pm sn}\, n^{\gamma_s}(1 + o(1))$$

satisfies

$$\|T^0_n\| \sim \max_{s\in S} \|K^0_s\|\, n^{2\,\mathrm{Re}\,\alpha} \quad \text{with} \quad \mathrm{Re}\,\alpha := \max_s \mathrm{Re}\,\alpha_s, \quad S = \{s : \mathrm{Re}\,\alpha_s = \mathrm{Re}\,\alpha\},$$

where $K^0_s$ is the operator whose kernel is $B^+_s(x-y)^{\gamma_s}$ for $x > y$ and $B^-_s(y-x)^{\gamma_s}$ for $x < y$. This is equivalent to saying that

$$\|T^0_n\| \sim \max_{r\in M} |b_r(t_r)|\, \|K_r\|\, n^{2\,\mathrm{Re}\,\alpha},$$

where the kernel of $K_r$ is $C^+_{\alpha_r,\beta_r}(x-y)^{2\,\alpha_r-1}$ for $x > y$ and $C^-_{\alpha_r,\beta_r}(y-x)^{2\,\alpha_r-1}$ for $x < y$. Since $\|T_n(f)\| = o(n^{2\,\mathrm{Re}\,\alpha})$, we obtain that $\|T_n\| \sim \|T^0_n\|$. □

**5. A particular singularity.** We finally embark on the case where

$$a(t) = |t - t_0|^{-2\alpha} b(t) \quad (t \in \mathbf{T})$$

with a real number $\alpha \in (0, 1/2)$ and a function $b \in L^\infty(\mathbf{T})$ that is continuous and nonzero at $t_0$. Theorem 3.3 gives

$$\|T_n(a)\| \sim \Gamma(1-2\alpha)\, \frac{\sin \pi\alpha}{\pi}\, \|K_\alpha\|\, |b(t_0)|\, n^{2\alpha},$$

where the kernel of $K_\alpha$ is $|x-y|^{2\alpha-1}$.

PROPOSITION 5.1.

$$\frac{1}{2\alpha}\left(\frac{2}{4\alpha+1}+2\,\frac{\Gamma(2\alpha+1)\Gamma(2\alpha+1)}{\Gamma(4\alpha+2)}\right)^{1/2}\le\|K_\alpha\|\le\frac{1}{\alpha}.$$

We may think of $K_\alpha$ as the compression to $L^2(0,1)$ of the convolution operator on $L^2(\mathbf{R})$ whose convolution kernel $\kappa(x)$ is $|x|^{2\alpha-1}$ for $|x|<1$ and 0 for $|x|>1$. As in the proof of Lemma 2.3 we therefore see that

$$\|K_\alpha\|\le\max_{\xi\in\mathbf{R}}|(F\kappa)(\xi)|\le\int_{\mathbf{R}}|\kappa(x)|\,dx=\int_{-1}^{1}|x|^{2\alpha-1}\,dx=\frac{1}{\alpha}.$$

Let $\mathbf{1}$ be the function which is identically 1 on $(0,1)$. Taking into account that

$$\|K_\alpha\|^2\ge\|K_\alpha\mathbf{1}\|^2/\|\mathbf{1}\|^2=\|K_\alpha\mathbf{1}\|^2\quad\text{and}\quad(K_\alpha\mathbf{1})(x)=\frac{1}{2\alpha}\,(x^{2\alpha}+(1-x)^{2\alpha}),$$

we obtain that $\|K_\alpha\|^2$ is greater than or equal to

$$\frac{1}{4\alpha^2}\int_0^1(x^{2\alpha}+(1-x)^{2\alpha})^2\,dx=\frac{1}{4\alpha^2}\left(\frac{2}{4\alpha+1}+2\,\frac{\Gamma(2\alpha+1)\Gamma(2\alpha+1)}{\Gamma(4\alpha+2)}\right).$$

This proves the lower bound for $\|K_\alpha\|$. □

COROLLARY 5.2. $\|K_\alpha\|\sim 1/\alpha$ $\alpha\to 0$ $\|K_\alpha\|\sim 1$ $\alpha\to 1/2$

By Proposition 5.1, $\alpha^2\,\|K_\alpha\|^2\le 1$ and

$$\liminf_{\alpha\to 0}\alpha^2\,\|K_\alpha\|^2\ge\frac{1}{4}\left(2+2\,\frac{\Gamma(1)\Gamma(1)}{\Gamma(2)}\right)=1,$$

which implies that $\alpha\,\|K_\alpha\|\to 1$ as $\alpha\to 0$. Thinking of $K_\alpha-K_{1/2}$ as the convolution operator with the convolution kernel $|x|^{2\alpha-1}-1$ for $|x|<1$ and 0 for $|x|>1$, we get

$$\|K_\alpha-K_{1/2}\|\le\int_{-1}^{1}(|x|^{2\alpha-1}-1)\,dx=\frac{1}{\alpha}-2=o(1)\ \text{ as }\ \alpha\to\frac{1}{2}.$$

Thus, $\|K_\alpha\|\to\|K_{1/2}\|$ as $\alpha\to 1/2$. Since $(K_{1/2}f)(x)=\int_0^1 f(y)\,dy$, it is easily seen that $\|K_{1/2}\|=1$. □

COROLLARY 5.3.

$$\begin{aligned}\Gamma(1-2\alpha)\,\frac{\sin\pi\alpha}{\pi}\,\|K_\alpha\|&\sim 1\qquad \alpha\to 0,\\ \Gamma(1-2\alpha)\,\frac{\sin\pi\alpha}{\pi}\,\|K_\alpha\|&\sim\frac{1}{2\pi(1/2-\alpha)}\qquad \alpha\to\frac{1}{2}.\end{aligned}$$

The asymptotics for $\alpha\to 0$ are immediate from Corollary 5.2. For $\alpha$ going to $1/2$, Corollary 5.2 and the formulas

$$\Gamma(1-2\alpha)\,\frac{\sin\pi\alpha}{\pi}\sim\frac{\Gamma(1-2\alpha)}{\pi}=\frac{1}{\sin 2\pi\alpha}\,\frac{1}{\Gamma(2\alpha)}\sim\frac{1}{2\pi(1/2-\alpha)}$$

yield the asserted asymptotics. □

**Acknowledgments.** The authors thank Fanny Godet and Bernhard Beckermann for drawing their attention to the problem considered in this paper. They are also greatly indebted to Torsten Ehrhardt for many valuable remarks. Jani Virtanen thanks the Department of Mathematics for its hospitality during his visit to TU Chemnitz in Spring 2006.

## REFERENCES

[1] A. Böttcher, *The Onsager formula, the Fisher–Hartwig conjecture, and their influence on research into Toeplitz operators,* J. Statist. Phys., 78 (Lars Onsager Festschrift) (1995), pp. 575–585.

[2] A. Böttcher and S. Grudsky, *On the condition numbers of large semi-definite Toeplitz matrices,* Linear Algebra Appl., 279 (1998), pp. 285–301.

[3] A. Böttcher and S. Grudsky, *Fejér means and norms of large Toeplitz matrices,* Acta Sci. Math. (Szeged), 69 (2003), pp. 889–900.

[4] A. Böttcher and B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices,* Universitext, Springer-Verlag, New York, 1999.

[5] A. Böttcher and B. Silbermann, *Analysis of Toeplitz Operators,* 2nd ed., Springer-Verlag, Berlin, Heidelberg, New York, 2006.

[6] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods,* 2nd ed., Springer-Verlag, New York, 1991.

[7] R. Dahlhaus, *Efficient parameter estimation for self-similar processes,* Ann. Statist., 17 (1989), pp. 1749–1766.

[8] P. Doukhan, G. Oppenheim, and M. S. Taqqu, eds., *Theory and Applications of Long-Range Dependence,* Birkhäuser Boston, Boston, 2003.

[9] T. Ehrhardt, *A status report on the asymptotic behavior of Toeplitz determinants with Fisher–Hartwig singularities,* in Recent Advances in Operator Theory (Groningen, 1998), Oper. Theory Adv. Appl. 124, Birkhäuser, Basel, 2001, pp. 217–241.

[10] R. Lewis and G. C. Reinsel, *Prediction of multivariate time series by autoregressive model fitting,* J. Multivariate Anal., 16 (1985), pp. 393–411.

[11] Y. Lu and C. M. Hurvich, *On the complexity of the preconditioned conjugate gradient algorithm for solving Toeplitz systems with a Fisher–Hartwig singularity,* SIAM J. Matrix Anal. Appl., 27 (2005), pp. 638–653.

[12] W. F. Trench, *Asymptotic distribution of the spectra of a class of generalized Kac–Murdock–Szegö matrices,* Linear Algebra Appl., 294 (1999), pp. 181–192.

[13] W. F. Trench, *Properties of some generalizations of Kac–Murdock–Szegö matrices,* in Structured Matrices in Mathematics, Computer Science and Engineering, II (Boulder, CO, 1999), Contemp. Math. 281, AMS, Providence, RI, 2001, pp. 233–245.

[14] W. F. Trench, *Spectral distribution of generalized Kac–Murdock–Szegö matrices,* Linear Algebra Appl., 347 (2002), pp. 251–273.

[15] E. E. Tyrtyshnikov and N. L. Zamarashkin, *Toeplitz eigenvalues for Radon measures,* Linear Algebra Appl., 343/344 (2002), pp. 345–354.

[16] H. Widom, *On the eigenvalues of certain Hermitian operators,* Trans. Amer. Math. Soc., 88 (1958), pp. 491–522.

[17] H. Widom, *Extreme eigenvalues of translation kernels,* Trans. Amer. Math. Soc., 100 (1961), pp. 252–262.

[18] H. Widom, *Extreme eigenvalues of $N$-dimensional convolution operators,* Trans. Amer. Math. Soc., 106 (1963), pp. 391–414.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 2, pp. 672–684

# CONSTRAINT-STYLE PRECONDITIONERS FOR REGULARIZED SADDLE POINT PROBLEMS*

H. S. DOLLAR[†]

**Abstract.** The problem of finding good preconditioners for the numerical solution of an important class of indefinite linear systems is considered. These systems are of a regularized saddle point structure $\left[\begin{smallmatrix} A & B^T \\ B & -C \end{smallmatrix}\right]\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right] = \left[\begin{smallmatrix} c \\ d \end{smallmatrix}\right]$, where $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times m}$ are symmetric and $B \in \mathbb{R}^{m\times n}$. In [*SIAM J. Matrix Anal. Appl.*, 21 (2000), pp. 1300–1317], Keller, Gould, and Wathen analyze the idea of using constraint preconditioners that have a specific 2 by 2 block structure for the case of $C$ being zero. We shall extend this idea by allowing the (2, 2) block to be symmetric and positive semidefinite. Results concerning the spectrum and form of the eigenvectors are presented, as are numerical results to validate our conclusions.

**Key words.** preconditioning, indefinite linear systems, Krylov subspace methods

**AMS subject classifications.** 65F10, 65F15, 65F50, 90C20

**DOI.** 10.1137/050626168

## 1. Introduction.

Recently, a large amount of work has been devoted to the problem of solving large linear systems in saddle point form. Such systems arise in a wide variety of technical and scientific applications. For example, interior point methods in both linear and nonlinear optimization require the solution of a sequence of systems in saddle point form [27]. Another popular field, which is a major source of saddle point problems, is that of mixed finite element methods in engineering fields; see [9] and [19, Chapters 7 and 9]. An excellent survey of numerical methods for algebraic saddle point problems has been written by Benzi, Golub, and Liesen [4].

We wish to find the solution of block $2 \times 2$ linear systems of the form

$$\underbrace{\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} x \\ y \end{bmatrix} = \underbrace{\begin{bmatrix} c \\ d \end{bmatrix}}_{b}, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times m}$ are symmetric and $B \in \mathbb{R}^{m\times n}$. We shall assume that $m \leq n$ and $\ker(C) \cap \ker(B^T) = \{0\}$, thus ensuring that $\mathcal{A}$ is nonsingular [4, Theorem 3.1]. If $A$ and $C$ are positive definite, then the matrix $\mathcal{A}$ is a permuted quasi-definite matrix [26]. Vanderbei has shown that quasi-definite matrices are strongly factorizable; i.e., a Cholesky-like factorization $LDL^T$ exists for any symmetric row and column permutation of the quasi-definite matrix [26]. The diagonal matrix has $n$ positive and $m$ negative pivots. However, we shall not confine ourselves to quasi-definite matrices.

It may be attractive to use iterative methods to solve systems such as (1.1), particularly for large $m$ and $n$. In particular, Krylov subspace methods might be used. It is often advantageous to use a preconditioner, $\mathcal{P}$, with such iterative methods. The preconditioner should reduce the number of iterations required for convergence but

---

[*] Received by the editors March 7, 2005; accepted for publication (in revised form) by V. Simoncini December 18, 2006; published electronically May 22, 2007.
http://www.siam.org/journals/simax/29-2/62616.html

[†] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxon OX11 0QX, UK (S.Dollar@rl.ac.uk).

not significantly increase the amount of computation required at each iteration [25, Chapter 13].

In section 2, we shall first review the well-known spectral properties of a technique commonly known as constraint preconditioning when $C = 0$ [14, 16]. For the case of $C = 0$, a constraint preconditioner exactly reproduces the (constraint) blocks $B$, $B^T$ and the $C = 0$ block. It is restrictive to assume that the matrix $C$ in the saddle point systems is always a zero matrix: a number of situations arise in which $C \neq 0$ [1, 15, 23]. In all these cases, $C$ is positive semidefinite, and hence we shall consider the idea of extending constraint preconditioners to the case of $C$ being positive semidefinite. In particular, the preconditioner will exactly reproduce the $B$, $B^T$ and $C$ blocks, while the $A$ block will be replaced by a symmetric block, which we refer to as $G$; this is considered in sections 3 and 4. Such a preconditioner has been considered before; for example, Perugia and Simoncini consider the case of $G$ being diagonal and positive definite [18], while $G$ is assumed to be nonsingular in [22] and positive definite in [3, 8, 24], but we show that these assumptions can be relaxed. In the past couple of years, the use of implicit factorization preconditioners has been proposed [7] with the aim of reducing the cost (both in CPU time and memory usage) of applying a preconditioner of the form suggested in this paper. However, such implicit factorization preconditioners will frequently generate a matrix $G$ which is symmetric and singular or indefinite, and thus the analysis of these preconditioners with such a $G$ is necessary.

**2. Constraint preconditioners.** Let us initially assume that $C = 0$. Lukšan and Vlček [17] and Keller, Gould, and Wathen [14] investigated the spectral properties of the resulting preconditioned system when we use a preconditioner of the form

$$\mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix}, \tag{2.1}$$

where $G$ is symmetric and approximates but (in general) is not the same as $A$. In [17], $G$ is additionally assumed to be positive definite. They were able to prove various results about the eigenvalues and eigenvectors for the preconditioned systems $\mathcal{P}^{-1}\mathcal{A}$, where $\mathcal{A}$ and $\mathcal{P}$ are defined in (1.1) and (2.1), respectively. $\mathcal{P}$ is called a [illegible]. The proof of the following theorem can be found in [14].

THEOREM 2.1. [illegible] $\mathcal{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ [illegible]

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix},$$

[illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $B \in \mathbb{R}^{m\times n}$ [illegible] $Z$ [illegible] $n \times (n-m)$ [illegible] $B$ [illegible] $\mathcal{A}$ [illegible]

$$\mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix},$$

[illegible] $G \in \mathbb{R}^{n\times n}$ [illegible] $B \in \mathbb{R}^{m\times n}$ [illegible] $\mathcal{P}^{-1}\mathcal{A}$ [illegible]

- [illegible] $1$ [illegible] $2m$
- $n - m$ [illegible] $\lambda$ [illegible] $Z^TAZx_z = \lambda Z^TGZx_z$.

If either $Z^TAZ$ or $Z^TGZ$ is positive definite, then the indefinite preconditioner $\mathcal{P}$ applied to the indefinite saddle point matrix $\mathcal{A}$ with $C = 0$ yields a preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ which has real eigenvalues [14]. If both $Z^TAZ$ and $Z^TGZ$ are positive definite, then we can use a projected preconditioned conjugate gradient method to find $x$ and $y$; see [12]. Results about the associated eigenvectors and the Krylov subspace dimension can also be found in [14].

## 3. Constraint preconditioners for the case of symmetric and positive definite *C*.

In this section, we shall assume that the matrix $C$ is symmetric and positive definite. The term [illegible] was used in [10] and [14] because the (1, 2) and (2, 1) matrix blocks of the preconditioner are exact representations of those in $\mathcal{A}$, where these blocks represent constraints. However, we also observe that the (2, 2) matrix block is an exact representation when $C = 0$. This motivates the generalization of the constraint preconditioner to take the form

$$\mathcal{P} = \begin{bmatrix} G & B^T \\ B & -C \end{bmatrix}, \tag{3.1}$$

where $G \in \mathbb{R}^{n\times n}$ approximates but is, in general, not the same as $A$.

We shall use the following assumptions in the theorems of this section.

A1 $C \in \mathbb{R}^{m\times m}$ is symmetric and positive definite.
A2 $A \in \mathbb{R}^{n\times n}$ is symmetric.
A3 $B \in \mathbb{R}^{m\times n}$ $(m < n)$.
A4 $G \in \mathbb{R}^{n\times n}$ is symmetric.
A5 $\mathcal{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ is as defined in (1.1).
A6 $\mathcal{P} \in \mathbb{R}^{(n+m)\times(n+m)}$ is as defined in (3.1).

In the next section, A1 will be relaxed.

THEOREM 3.1. [illegible] A1–A6 [illegible] $\mathcal{P}^{-1}\mathcal{A}$ [illegible]

- [illegible] 1 [illegible] $m$,
- $n$ [illegible]

$$\left(A + B^TC^{-1}B\right)x = \lambda\left(G + B^TC^{-1}B\right)x.$$

[illegible]. The eigenvalues of the preconditioned coefficient matrix $\mathcal{P}^{-1}\mathcal{A}$ may be derived by considering the generalized eigenvalue problem

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} G & B^T \\ B & -C \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{3.2}$$

Expanding this out, we obtain

$$Ax + B^Ty = \lambda Gx + \lambda B^Ty \tag{3.3}$$

and

$$Bx - Cy = \lambda Bx - \lambda Cy. \tag{3.4}$$

Equation (3.4) implies that either $\lambda = 1$ or $Bx - Cy = 0$. If the former holds, then (3.3) becomes

$$Ax = Gx. \tag{3.5}$$

Equation (3.5) is trivially satisfied by $x = 0$, and hence there are $m$ linearly independent eigenvectors of the form $\begin{bmatrix} 0^T & y^T \end{bmatrix}$ associated with the unit eigenvalue. If there exist any $x \neq 0$ which satisfy (3.5), then there will be $i$ $(0 \leq i \leq n)$ linearly independent eigenvectors of the form $\begin{bmatrix} x^T & y^T \end{bmatrix}$, where the components $x$ arise from the generalized eigenvalue problem $Ax = Gx$.

If $\lambda \neq 1$, then (3.4) implies that

$$y = C^{-1}Bx.$$

Substituting this into (3.3) yields the generalized eigenvalue problem

$$\left(A + B^TC^{-1}B\right)x = \lambda\left(G + B^TC^{-1}B\right)x. \tag{3.6}$$

Thus, the nonunit eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ are defined as the nonunit eigenvalues of (3.6). Noting that if (3.6) has any unit eigenvalues, then the values of $x(\neq 0)$ which satisfy this are exactly those which arise from the generalized eigenvalue problem $Ax = Gx$, we complete our proof. □

Theorem 3.1 generalizes the results of [8, Theorem 1] by removing the assumption that $G$ is positive definite. If $A + B^TC^{-1}B$ or $G + B^TC^{-1}B$ is positive definite, then the preconditioned system has real eigenvalues. If both $A + B^TC^{-1}B$ and $G + B^TC^{-1}B$ are positive definite, then we can apply a projected preconditioned conjugate gradient method to find $x$ and $y$ [7, 11]. We also observe that if $C$ has a small 2-norm, $\|A\|_2 = \mathcal{O}(1)$ and $\|G\|_2 = \mathcal{O}(1)$, then the $B^TC^{-1}B$ terms will dominate the generalized eigenvalue problem (3.6) for $Bx \neq 0$, and hence there will be at least $m$ further eigenvalues clustered about 1 for $\|C\|_2 \ll 1$. This additional clustering of part of the spectrum of $\mathcal{P}^{-1}\mathcal{A}$ will often translate into a speeding up of the convergence of a selected Krylov subspace method [2, section 1.3].

THEOREM 3.2. [illegible] A1–A6 [illegible] $G + B^TC^{-1}B$ [illegible] $\mathcal{P}^{-1}\mathcal{A}$ [illegible] $n+m$ [illegible] 3.1 [illegible] $m+i+j$ [illegible]

- $m$ [illegible] $\begin{bmatrix} 0^T & y^T \end{bmatrix}$ [illegible] $\lambda = 1$,
- $i$ $(0 \leq i \leq n)$ [illegible] $\begin{bmatrix} x^T & y^T \end{bmatrix}$ [illegible] $Ax = Gx$ [illegible] $i$ [illegible] $x$ [illegible] $\lambda = 1$,
- $j$ $(0 \leq j \leq n)$ [illegible] $\begin{bmatrix} x^T & y^T \end{bmatrix}$ [illegible] $\lambda \neq 1$.

[illegible]. The form of the eigenvectors follows directly from the proof of Theorem 3.1. It remains for us to show that the $m + i + j$ eigenvectors are linearly independent; that is, we need to show that

$$\begin{bmatrix} 0 & \cdots & 0 \\ y_1^{(1)} & \cdots & y_m^{(1)} \end{bmatrix} \begin{bmatrix} a_1^{(1)} \\ \vdots \\ a_m^{(1)} \end{bmatrix} + \begin{bmatrix} x_1^{(2)} & \cdots & x_i^{(2)} \\ y_1^{(2)} & \cdots & y_i^{(2)} \end{bmatrix} \begin{bmatrix} a_1^{(2)} \\ \vdots \\ a_i^{(2)} \end{bmatrix}$$
$$+ \begin{bmatrix} x_1^{(3)} & \cdots & x_j^{(3)} \\ y_1^{(3)} & \cdots & y_j^{(3)} \end{bmatrix} \begin{bmatrix} a_1^{(3)} \\ \vdots \\ a_j^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.7}$$

implies that the vectors $a^{(k)}$ $(k = 1, 2, 3)$ are zero vectors. Multiplying (3.7) by $\mathcal{P}^{-1}\mathcal{A}$, and recalling that in (3.7) the first matrix arises from the case $\lambda_k = 1$ $(k = 1, \ldots, m)$, the second matrix from the case $\lambda_k = 1$ $(k = 1, \ldots, i)$, and the last matrix from $\lambda_k \neq 1$ $(k = 1, \ldots, j)$, gives

$$\begin{bmatrix} 0 & \cdots & 0 \\ y_1^{(1)} & \cdots & y_m^{(1)} \end{bmatrix} \begin{bmatrix} a_1^{(1)} \\ \vdots \\ a_m^{(1)} \end{bmatrix} + \begin{bmatrix} x_1^{(2)} & \cdots & x_i^{(2)} \\ y_1^{(2)} & \cdots & y_i^{(2)} \end{bmatrix} \begin{bmatrix} a_1^{(2)} \\ \vdots \\ a_i^{(2)} \end{bmatrix}$$

$$+ \begin{bmatrix} x_1^{(3)} & \cdots & x_j^{(3)} \\ y_1^{(3)} & \cdots & y_j^{(3)} \end{bmatrix} \begin{bmatrix} \lambda_1 a_1^{(3)} \\ \vdots \\ \lambda_j a_j^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3.8}$$

Subtracting (3.7) from (3.8), we obtain

$$\begin{bmatrix} x_1^{(3)} & \cdots & x_j^{(3)} \\ y_1^{(3)} & \cdots & y_j^{(3)} \end{bmatrix} \begin{bmatrix} (\lambda_1 - 1) a_1^{(3)} \\ \vdots \\ (\lambda_j - 1) a_j^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The assumption that $G + B^T C^{-1} B$ is positive definite implies that $x_k^{(3)}$ $(k = 1, \ldots, j)$ are linearly independent and thus that $(\lambda_k - 1) a_1^{(3)} = 0$ $(k = 1, \ldots, j)$. The eigenvalues $\lambda_k$ $(k = 1, \ldots, j)$ are nonunit, which implies that $a_k^{(3)} = 0$ $(k = 1, \ldots, j)$. We also have linear independence of $x_k^{(2)}$ $(k = 1, \ldots, i)$, and thus $a_k^{(2)} = 0$ $(k = 1, \ldots, i)$. Equation (3.7) simplifies to

$$\begin{bmatrix} 0 & \cdots & 0 \\ y_1^{(1)} & \cdots & y_m^{(1)} \end{bmatrix} \begin{bmatrix} a_1^{(1)} \\ \vdots \\ a_m^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

However, $y_k^{(1)}$ $(k = 1, \ldots, m)$ are linearly independent, and thus $a_k^{(1)} = 0$ $(k = 1, \ldots, m)$. □

Krylov subspace theory states that iteration with any method with an optimality property, e.g., GMRES [21], will terminate when the degree of the minimal polynomial is attained. This is also true of some other (nonoptimal) practical iteration methods such as BiCGTAB as long as failure (for example, through irregular convergence [25, Chapter 8]) does not occur. In particular, the degree of the minimal polynomial is equal to the dimension of the corresponding Krylov subspace $\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right)$ (for general $b$) [20, Proposition 6.1], where

$$\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right) = \operatorname{span}\{b, \mathcal{P}^{-1}\mathcal{A}b, (\mathcal{P}^{-1}\mathcal{A})^2 b, \ldots, (\mathcal{P}^{-1}\mathcal{A})^{n+m-1} b\}.$$

THEOREM 3.3. [illegible] A1–A6 [illegible] $G + B^T C^{-1} B$ [illegible] $\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right)$ [illegible] $\min\{n+2, n+m\}$.

[illegible] As in the proof of Theorem 3.1, the generalized eigenvalue problem is

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} G & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{3.9}$$

Suppose that the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ takes the form

$$\mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} \Theta_1 & \Theta_3 \\ \Theta_2 & \Theta_4 \end{bmatrix}, \tag{3.10}$$

where $\Theta_1 \in \mathbb{R}^{n\times n}$, $\Theta_2 \in \mathbb{R}^{m\times n}$, $\Theta_3 \in \mathbb{R}^{n\times m}$, and $\Theta_4 \in \mathbb{R}^{m\times m}$. It is straightforward to show that $\Theta_3 = 0$ and $\Theta_4 = I$. The precise forms of $\Theta_1$ and $\Theta_2$ are irrelevant for the argument that follows.

From the earlier eigenvalue derivation, it is evident that the characteristic polynomial of the preconditioned linear system (3.10) is

$$\left(\mathcal{P}^{-1}\mathcal{A} - I\right)^m \prod_{i=1}^{n} \left(\mathcal{P}^{-1}\mathcal{A} - \lambda_i I\right).$$

In order to prove the upper bound on the Krylov subspace dimension, we need to show that the order of the minimal polynomial is less than or equal to $\min\{n+2, n+m\}$. Expanding the polynomial $\left(\mathcal{P}^{-1}\mathcal{A} - I\right) \prod_{i=1}^{n} \left(\mathcal{P}^{-1}\mathcal{A} - \lambda_i I\right)$ of degree $n+1$, we obtain

$$\left[\begin{array}{cc} (\Theta_1 - I)\prod_{i=1}^{n}(\Theta_1 - \lambda_i I) & 0 \\ \Theta_2 \prod_{i=1}^{n}(\Theta_1 - \lambda_i I) & 0 \end{array}\right].$$

Since $\Theta_1$ has a full set of linearly independent eigenvectors, $\Theta_1$ is diagonalizable. Hence,

$$(\Theta_1 - I)\prod_{i=1}^{n}(\Theta_1 - \lambda_i I) = 0.$$

We therefore obtain

$$\left(\mathcal{P}^{-1}\mathcal{A} - I\right) \prod_{i=1}^{n} \left(\mathcal{P}^{-1}\mathcal{A} - \lambda_i I\right) = \left[\begin{array}{cc} 0 & 0 \\ \Theta_2 \prod_{i=1}^{n}(\Theta_1 - \lambda_i I) & 0 \end{array}\right]. \tag{3.11}$$

If $\Theta_2 \prod_{i=1}^{n}(\Theta_1 - \lambda_i I) = 0$, then the order of the minimal polynomial of $\mathcal{P}^{-1}\mathcal{A}$ is less than or equal to $\min\{n+1, n+m\}$. If $\Theta_2 \prod_{i=1}^{n}(\Theta_1 - \lambda_i I) \neq 0$, then the dimension of $\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right)$ is at most $\min\{n+2, n+m\}$ since multiplication of (3.11) by another factor $\left(\mathcal{P}^{-1}\mathcal{A} - I\right)$ gives the zero matrix. ☐

Theorem 3.3 tells us that with preconditioner

$$\mathcal{P} = \left[\begin{array}{cc} G & B^T \\ B & -C \end{array}\right]$$

for

$$\mathcal{A} = \left[\begin{array}{cc} A & B^T \\ B & -C \end{array}\right]$$

the dimension of the Krylov subspace is no greater than $\min\{n+2, n+m\}$ under appropriate assumptions. Hence, termination (in exact arithmetic) is guaranteed in a number of iterations smaller than this.

**4. Constraint preconditioners for the case of symmetric and positive semi-definite *C*.** We shall relax assumption A1 and instead make the following assumptions in the theorems of this section:

B1 $C \in \mathbb{R}^{m\times m}$ is symmetric and positive semidefinite, and has rank $p$, where $0 < p < m$.

B2 $\ker(C) \cap \ker(B^T) = \{0\}$.

- B3 $C$ is factored as $C = EDE^T$, where $E \in \mathbb{R}^{m\times p}$, and $D \in \mathbb{R}^{p\times p}$ is symmetric and positive definite.
- B4 The matrix $F \in \mathbb{R}^{m\times(m-p)}$ is such that its columns span the nullspace of $C$.
- B5 $\begin{bmatrix} E & F \end{bmatrix} \in \mathbb{R}^{m\times m}$ is orthogonal.
- B6 The columns of $N \in \mathbb{R}^{n\times(n-m+p)}$ span the nullspace of $F^TB$.

Observe that assumption B2 implies that $F^TB$ has full rank $m-p$ : if $Cx = 0$, then we can write $x = Fy$ for some vector $y \in \mathbb{R}^{m-p}$. If also $B^Tx = 0$, then substituting into $x = Fy$ we obtain $B^TFy = 0$. Assumption B2 implies that $B^TFy = 0$ if and only if $y = 0$, and hence $F^TB$ has full rank $m-p$.

The exact form of the factorization of $C$ in B3 is clearly not relevant and, also, clearly not unique—a spectral decomposition is a possibility.

THEOREM 4.1. [illegible] A2–A6 [illegible] B1–B6 [illegible] $\mathcal{P}^{-1}\mathcal{A}$ [illegible]

- [illegible] 1 [illegible] $2m-p$,
- $n-m+p$ [illegible]

$$N^T\left(A + B^TED^{-1}E^TB\right)Nz = \lambda N^T\left(G + B^TED^{-1}E^TB\right)Nz.$$

[illegible]

[illegible]. Any $y \in \mathbb{R}^m$ can be written as $y = Ey_e + Fy_f$. Substituting this into the generalized eigenvalue problem (3.2) and premultiplying by

$$\begin{bmatrix} I & 0 \\ 0 & E^T \\ 0 & F^T \end{bmatrix},$$

we obtain

$$\left[\begin{array}{cc|c} A & B^TE & B^TF \\ E^TB & -D & 0 \\ \hline F^TB & 0 & 0 \end{array}\right]\begin{bmatrix} x \\ y_e \\ y_f \end{bmatrix} = \lambda\left[\begin{array}{cc|c} G & B^TE & B^TF \\ E^TB & -D & 0 \\ \hline F^TB & 0 & 0 \end{array}\right]\begin{bmatrix} x \\ y_e \\ y_f \end{bmatrix}. \tag{4.1}$$

Noting that the (3, 3) block has dimension $(m-p)\times(m-p)$ and is a zero matrix in both coefficient matrices, we can apply Theorem 2.1 from [14] to obtain that $\mathcal{P}^{-1}\mathcal{A}$ has

- an eigenvalue at 1 with multiplicity $2(m-p)$,
- $n-m+2p$ eigenvalues which are defined by the generalized eigenvalue problem

$$\overline{N}^T\begin{bmatrix} A & B^TE \\ E^TB & -D \end{bmatrix}\overline{N}w_n = \lambda\overline{N}^T\begin{bmatrix} G & B^TE \\ E^TB & -D \end{bmatrix}\overline{N}w_n, \tag{4.2}$$

where $\overline{N}$ is an $(n+p)\times(n-m+2p)$ basis for the nullspace of $\begin{bmatrix} F^TB & 0 \end{bmatrix} \in \mathbb{R}^{(m-p)\times(n+p)}$, and

$$\begin{bmatrix} x \\ y_e \end{bmatrix}^T = \overline{N}w_n + \begin{bmatrix} B^TF \\ 0 \end{bmatrix}w_b.$$

Letting $\overline{N} = \left[\begin{smallmatrix} N & 0 \\ 0 & I \end{smallmatrix}\right]$, then (4.2) becomes

$$\begin{bmatrix} N^TAN & N^TB^TE \\ E^TBN & -D \end{bmatrix}\begin{bmatrix} w_{n1} \\ w_{n2} \end{bmatrix} = \lambda\begin{bmatrix} N^TGN & N^TB^TE \\ E^TBN & -D \end{bmatrix}\begin{bmatrix} w_{n1} \\ w_{n2} \end{bmatrix}. \tag{4.3}$$

This generalized eigenvalue problem is exactly that of the form considered in Theorem 3.1, and so (4.3) has an eigenvalue at 1 with multiplicity $p$, and the remaining eigenvalues are defined by the generalized eigenvalue problem

$$N^T\left(A+B^TED^{-1}E^TB\right)Nw_{n1}=\lambda N^T\left(G+B^TED^{-1}E^TB\right)Nw_{n1}. \tag{4.4}$$

Hence, $\mathcal{P}^{-1}\mathcal{A}$ has an eigenvalue at 1 with multiplicity $2m-p$, and the other eigenvalues are defined by the generalized eigenvalue problem (4.4). ☐

Weaker forms of Theorem 4.1 can be found in [3, section 3.7] and [18, Proposition 5] for the case where $G$ is assumed to be symmetric and positive definite (and diagonal in [18]). We have relaxed this assumption to $G$ being symmetric and also increased the lower bound on the number of unit eigenvalues from $m$ to $2m-p$.

As for the cases $C=0$ and $C$ nonsingular, we are able to obtain conditions which guarantee that the eigenvalues are real and for which a projected preconditioned conjugate gradient method could be applied to find $x$ and $y$; respectively, these conditions are

- either $N^T\left(A+B^TED^{-1}E^TB\right)N$ or $N^T\left(G+B^TED^{-1}E^TB\right)N$ is positive definite,
- both $N^T\left(A+B^TED^{-1}E^TB\right)N$ and $N^T\left(G+B^TED^{-1}E^TB\right)N$ are positive definite.

Interestingly, the projected preconditioned conjugate gradient method is also derived by the use of a factorization of $C$ as in assumption B3; transformations are then used to remove the requirement of needing to factorize $C$ [7]. Additionally, in [7] the authors show that it can be easy to establish that $N^T\left(G+B^TED^{-1}E^TB\right)N$ is symmetric and positive definite through the use of implicit factorization constraint preconditioners: we emphasize that $G$ is often singular or indefinite in these cases.

Similarly to the case $p=m$, if $C$ has a small 2-norm, $\|A\|=\mathcal{O}(1)$ and $\|G\|=\mathcal{O}(1)$, then the $N^TB^TED^{-1}E^TBN$ terms will dominate the generalized eigenvalue problem (4.4) for $E^TBNw_{n1}\neq 0$ and hence there will be at least $p$ further eigenvalues clustered about 1 when $\|C\|_2 \ll 1$.

THEOREM 4.2. A2–A6 B1–B6 $G+B^TED^{-1}E^TB$ $\mathcal{P}^{-1}\mathcal{A}$ $n+m$ 3.1 $m+i+j$

- $m$ $\left[\begin{array}{cc} 0^T & y^T \end{array}\right]$ $\lambda=1$,
- $i$ $(0\le i\le n)$ $\left[\begin{array}{cc} x^T & y^T \end{array}\right]$ $Ax=Gx$ $i$ $x$ $\lambda=1$,
- $j$ $(0\le j\le n)$ $\left[\begin{array}{cc} x^T & y^T \end{array}\right]$ $\lambda\neq 1$.

The proof of the form and linear independence of the $m+i+j$ eigenvectors is obtained in a similar manner to the proof of Theorem 3.2. ☐

A weaker form of Theorem 4.2 can be found in [3]: this corresponds to the case of $G$ being symmetric and positive definite.

To show that both the lower and upper bounds on the number of linearly independent eigenvectors can be attained, we need only consider variations on Examples 2.5 and 2.6 from [14].

4.1 (minimum bound). Consider the matrices

$$\mathcal{A}=\begin{bmatrix}1&2&0&1\\2&2&1&0\\0&1&0&0\\1&0&0&-1\end{bmatrix},\quad \mathcal{P}=\begin{bmatrix}1&3&0&1\\3&4&1&0\\0&1&0&0\\1&0&0&-1\end{bmatrix}$$

such that $m = 2$, $n = 2$, $p = 1$, and $G$ is indefinite. The preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ has an eigenvalue at 1 with multiplicity 4 but only two linearly independent eigenvectors which arise from the first case of Theorem 4.2. These eigenvectors may be taken to be $\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$ and $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$.

[illegible] 4.2 (maximum bound). Let $\mathcal{A} \in \mathbb{R}^{4\times 4}$ be as defined in Example 4.1, but assume that $G = A$. The preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ has an eigenvalue at 1 with multiplicity 4 and clearly a complete set of eigenvectors. These may be taken to be the columns of the identity matrix.

The linear independence of the $m + i + j$ eigenvectors allows us to obtain an upper bound on the dimension of the Krylov subspace $\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right)$.

THEOREM 4.3. [illegible] A2–A6, B1–B6 [illegible] $G + B^TED^{-1}E^TB$ [illegible] $\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right)$ [illegible] $\min\{n - m + p + 2, n + m\}$.

[illegible]. As in the proof of Theorem 3.3, the preconditioned matrix $\mathcal{P}^{-1}\mathcal{A}$ takes the form

$$\mathcal{P}^{-1}\mathcal{A} = \begin{bmatrix} \Theta_1 & 0 \\ \Theta_2 & I \end{bmatrix}, \tag{4.5}$$

where $\Theta_1 \in \mathbb{R}^{n\times n}$, and $\Theta_2 \in \mathbb{R}^{m\times n}$. The precise forms of $\Theta_1$ and $\Theta_2$ are irrelevant for the argument that follows.

From the earlier eigenvalue derivation, it is evident that the characteristic polynomial of the preconditioned linear system (4.5) is

$$\left(\mathcal{P}^{-1}\mathcal{A} - I\right)^{2m-p} \prod_{i=1}^{n-m+p} \left(\mathcal{P}^{-1}\mathcal{A} - \lambda_i I\right).$$

In order to prove the upper bound on the Krylov subspace dimension, we need to show that the order of the minimal polynomial is less than or equal to $\min\{n - m + p + 2, n + m\}$. Expanding the polynomial $\left(\mathcal{P}^{-1}\mathcal{A} - I\right) \prod_{i=1}^{n-m+p} \left(\mathcal{P}^{-1}\mathcal{A} - \lambda_i I\right)$ of degree $n + 1$, we obtain

$$\begin{bmatrix} (\Theta_1 - I) \prod_{i=1}^{n-m+p} (\Theta_1 - \lambda_i I) & 0 \\ \Theta_2 \prod_{i=1}^{n-m+p} (\Theta_1 - \lambda_i I) & 0 \end{bmatrix}.$$

Since $G + B^TED^{-1}E^TB$ is positive definite, $\Theta_1$ has a full set of linearly independent eigenvectors and is diagonalizable. Hence, $(\Theta_1 - I) \prod_{i=1}^{n-m+p} (\Theta_1 - \lambda_i I) = 0$. We therefore obtain

$$\left(\mathcal{P}^{-1}\mathcal{A} - I\right) \prod_{i=1}^{n-m+p} \left(\mathcal{P}^{-1}\mathcal{A} - \lambda_i I\right) = \begin{bmatrix} 0 & 0 \\ \Theta_2 \prod_{i=1}^{n-m+p} (\Theta_1 - \lambda_i I) & 0 \end{bmatrix}. \tag{4.6}$$

If $\Theta_2 \prod_{i=1}^{n-m+p} (\Theta_1 - \lambda_i I) = 0$, then the order of the minimal polynomial of $\mathcal{P}^{-1}\mathcal{A}$ is less than or equal to $\min\{n - m + p + 1, n + m\}$. If $\Theta_2 \prod_{i=1}^{n-m+p} (\Theta_1 - \lambda_i I) = 0$, then the dimension of $\mathcal{K}\left(\mathcal{P}^{-1}\mathcal{A}, b\right)$ is at most $\min\{n-m+p+2, n+m\}$ since multiplication of (4.6) by another factor $\left(\mathcal{P}^{-1}\mathcal{A} - I\right)$ gives the zero matrix. □

Thus, in exact arithmetic, iteration with any method with an optimality condition will terminate in at most $\min\{n - m + p + 2, n + m\}$ iterations (in practice, exact arithmetic is not available, and hence this theoretical bound may be exceeded). We observe that if $p = m$, then Theorem 4.3 gives the same bound on the Krylov subspace dimension as that in Theorem 3.3, and if $p = 0$, then we obtain the results of [14].

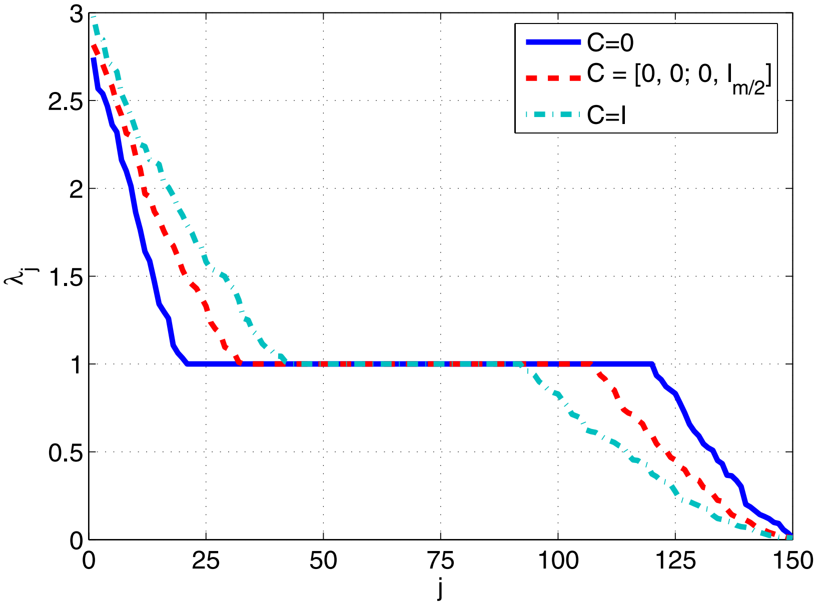


FIG. 5.1. *Distribution of the eigenvalues of* $\mathcal{P}^{-1}\mathcal{A}$ *for the CVXQP1_S problem* $(m = 50, n = 100)$ *with* $C = 0$, $C = [0, 0; 0, I_{m/2}]$, *and* $C = I$. *The eigenvalues are sorted such that* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n+m}$.

## 5. Numerical results.

The CUTEr test set [13] provides a set of quadratic programming problems. We shall use a problem from this set to illustrate how changing the rank of $C$ affects the multiplicity of the unit eigenvalues and the termination of GMRES. All tests were performed in MATLAB 7.01.

The CVXQP1_S problem from the CUTEr test set is small with $n = 100$ and $m = 50$. It is a convex quadratic program whose constraints are linear; it is a purely academic problem which has been constructed specifically for test problems. “Barrier” penalty terms (in this case 1.1) are added to the diagonal of $A$ to simulate systems that might arise during an iteration of an interior point method for such problems. We shall set $G = \mathrm{diag}(A)$, $C = \mathrm{diag}(0, \ldots, 0, 1, \ldots, 1)$ and vary the number of zeros on the diagonal of $C$ so as to change its rank.

In Figure 5.1, we illustrate the change in the eigenvalues of the preconditioned system $\mathcal{P}^{-1}\mathcal{A}$ for three different choices of $C$. The eigenvalues are sorted so that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{n+m}.$$

When $C = 0$, we expect there to be at least $2m$ unit eigenvalues [14]. We observe that our example has exactly $2m$ eigenvalues at 1. From Theorem 3.1, when $C = I$, there will be at least $m$ unit eigenvalues. Our example has exactly $m$ unit eigenvalues (Figure 5.1).

When $C$ has rank $\frac{m}{2}$, then the preconditioned system $\mathcal{P}^{-1}\mathcal{A}$ has at least $\frac{3m}{2}$ unit eigenvalues according to Theorem 4.1. Once again, the number of unit eigenvalues for our example is exactly the lower bound given by the theorem.

Now suppose that we use (full) GMRES preconditioned by our extended constraint preconditioner with $G = \mathrm{diag}(A)$ and vary the rank of $C$ by changing the

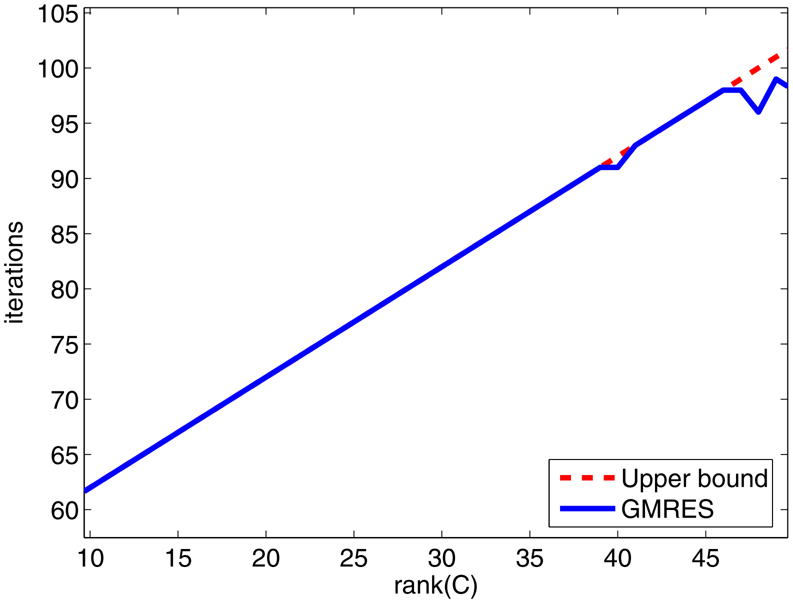


Fig. 5.2. *Comparison of upper bound on the Krylov subspace dimension and the number of iterations required to reduce the residual by* $10^{-12}$.

number of 1's along the diagonal of $C$ (all other entries are zero). Figure 5.2 shows that with this example and choice of $G$ there is a strong correlation between the upper bound on the Krylov subspace dimension and the number of iterations required to reduce the residual by at least a factor of $10^{-12}$. This has been chosen as an extreme example, and the number of iterations is often a lot lower than the upper bound on the Krylov subspace dimension. A comprehensive comparison (taking into account both CPU times and the number of iterations) for these preconditioners can be found in [7]: this study reveals the possible advantages of choosing $G$ to be singular or indefinite.

**6. Conclusions.** In this paper, we investigated a class of preconditioners for regularized saddle point matrix systems that incorporate the (1, 2), (2, 1), and (2, 2) blocks of the original matrix. We showed that the inclusion of these blocks in the preconditioner clusters at least $2m-p$ eigenvalues at 1, regardless of the structure of $G$. However, the standard convergence theory for Krylov subspace methods is not readily applicable because, in general, $\mathcal{P}^{-1}\mathcal{A}$ does not have a complete set of linearly independent eigenvectors. Using a minimal polynomial argument, we found a general (sharp) upper bound on the number of iterations required to solve linear systems of the form (1.1).

To confirm the analytical results of this paper, we used a subset of problems from the CUTEr test set. We used the CVXQP1_S problem and varied the rank of $C$ to confirm the lower bound on the number of unit eigenvalues and the upper bound on the Krylov subspace dimension.

We have assumed that the submatrices $B$, $B^T$ and $-C$ in (1.1) are exactly reproduced in the preconditioner. For truly large-scale problems, this will be unrealistic

[5, 6, 18], but the theorems in this paper may still be of some interest in the inexact setting as a guide for choosing preconditioners. We wish to investigate this possibility in our future work.

**Acknowledgments.** The author would like to thank Nick Gould, Wil Schilders, Andy Wathen, and the referees for their helpful input during the process of the work.

## REFERENCES

[1] A. Altman and J. Gondzio, *Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization*, Optim. Methods Softw., 11/12 (1999), pp. 275–302.

[2] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Classics Appl. Math. 35, SIAM, Philadelphia, 2001. Reprint of the 1984 original.

[3] O. Axelsson and M. Neytcheva, *Preconditioning methods for linear systems arising in constrained optimization problems*, Numer. Linear Algebra Appl., 10 (2003), pp. 3–31.

[4] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[5] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli, *Inexact constraint preconditioners for linear systems arising in interior point methods*, Comput. Optim. Appl., 36 (2007), pp. 137–147.

[6] G. Biros and O. Ghattas, *A Lagrange-Newton-Krylov-Schur method for PDE-constrained optimization*, SIAG/Optimization Views-and-News, 11 (2000), pp. 12–18.

[7] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 170–189.

[8] C. Durazzi and V. Ruggiero, *Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems*, Numer. Linear Algebra Appl., 10 (2003), pp. 673–688.

[9] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, UK, 2005.

[10] R. E. Ewing, R. D. Lazarov, P. Lu, and P. S. Vassilevski, *Preconditioning indefinite systems arising from mixed finite element discretization of second-order elliptic problems*, in Preconditioned Conjugate Gradient Methods (Nijmegen, 1989), Lecture Notes in Math. 1457, Springer, Berlin, 1990, pp. 28–43.

[11] N. I. M. Gould, *Iterative methods for ill-conditioned linear systems from optimization*, in Nonlinear Optimization and Related Topics, G. DiPillo and F. Giannessi, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 123–142.

[12] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.

[13] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr and SifDec: a constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.

[14] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[15] A. Klawonn, *An optimal preconditioner for a class of saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 540–552.

[16] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 219–247.

[17] L. Lukšan and J. Vlček, *Interior-point method for non-linear non-convex optimization*, Numer. Linear Algebra Appl. 11 (2004), pp. 431–453.

[18] I. Perugia and V. Simoncini, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.

[19] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer Ser. Comput. Math. 23, Springer, Berlin, 1994.

[20] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.

[21] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[22] C. Siefert and E. de Sturler, *Preconditioners for generalized saddle-point problems*, SIAM J. Numer. Anal., 44 (2006), pp. 1275–1296.

[23] D. Silvester and A. Wathen, *Fast iterative solution of stabilised Stokes systems Part* II. *Using general block preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.

[24] K.-C. Toh, K.-K. Phoon, and S.-H. Chan, *Block preconditioners for symmetric indefinite linear systems*, Internat. J. Numer. Methods Engrg., 60 (2004), pp. 1361–1381.

[25] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monogr. Appl. Comput. Math. 13, Cambridge University Press, Cambridge, UK, 2003.

[26] R. J. Vanderbei, *Symmetric quasidefinite matrices*, SIAM J. Optim., 5 (1995), pp. 100–113.

[27] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 2, pp. 685–697

# THE A PRIORI TAN Θ THEOREM FOR EIGENVECTORS*

SERGIO ALBEVERIO[†], ALEXANDER K. MOTOVILOV[‡], AND ALEXEI V. SELIN[§]

**Abstract.** Let $A$ be a self-adjoint operator on a Hilbert space $\mathfrak{H}$. Assume that the spectrum of $A$ consists of two disjoint components $\sigma_0$ and $\sigma_1$ such that the convex hull of the set $\sigma_0$ does not intersect the set $\sigma_1$. Let $V$ be a bounded self-adjoint operator on $\mathfrak{H}$ off-diagonal with respect to the orthogonal decomposition $\mathfrak{H} = \mathfrak{H}_0 \oplus \mathfrak{H}_1$, where $\mathfrak{H}_0$ and $\mathfrak{H}_1$ are the spectral subspaces of $A$ associated with the spectral sets $\sigma_0$ and $\sigma_1$, respectively. It is known that if $\|V\| < \sqrt{2}d$, where $d = \operatorname{dist}(\sigma_0, \sigma_1) > 0$, then the perturbation $V$ does not close the gaps between $\sigma_0$ and $\sigma_1$. Assuming that $f$ is an eigenvector of the perturbed operator $A+V$ associated with its eigenvalue in the interval $(\min(\sigma_0)-d, \max(\sigma_0)+d)$, we prove that under the condition $\|V\| < \sqrt{2}d$ the (acute) angle $\theta$ between $f$ and the orthogonal projection of $f$ onto $\mathfrak{H}_0$ satisfies the bound $\tan\theta \le \frac{\|V\|}{d}$, and this bound is sharp.

**Key words.** perturbation problem, spectral subspaces, perturbation of eigenvectors, $\tan\theta$ theorem

**AMS subject classifications.** 47A55, 47B25

**DOI.** 10.1137/06065667X

## 1. Introduction.

Given a self-adjoint operator $A$ on a Hilbert space $\mathfrak{H}$, assume that $\sigma_0$ is an isolated part of its spectrum, that is,

$$d = \operatorname{dist}(\sigma_0, \sigma_1) > 0, \tag{1.1}$$

where $\sigma_1 = \operatorname{spec}(A) \setminus \sigma_0$ is the rest of the spectrum of $A$. In this case, we say that there are open gaps between the sets $\sigma_0$ and $\sigma_1$. It is well known (see, e.g., [10, section 135]) that a sufficiently small self-adjoint perturbation $V$ of $A$ does not close these gaps, which allows one to think of the corresponding disjoint spectral components $\sigma_0'$ and $\sigma_1'$ of the perturbed operator $L = A + V$ as a result of the perturbation of the spectral sets $\sigma_0$ and $\sigma_1$, respectively.

Assuming (1.1), in this note we are concerned with perturbations $V$ that are off-diagonal with respect to the partition $\operatorname{spec}(A) = \sigma_0 \cup \sigma_1$, i.e., with perturbations that anticommute with the difference $\mathsf{E}_A(\sigma_0) - \mathsf{E}_A(\sigma_1)$ of the spectral projections $\mathsf{E}_A(\sigma_0)$ and $\mathsf{E}_A(\sigma_1)$ associated with the spectral sets $\sigma_0$ and $\sigma_1$, respectively. In general, it is known (see [6, Theorem 1]) that such perturbations do not close the gaps between the sets $\sigma_0$ and $\sigma_1$ (which means that the inequality $\operatorname{dist}(\sigma_0', \sigma_1') > 0$ holds) whenever

$$\|V\| < \frac{\sqrt{3}}{2}d. \tag{1.2}$$

---

*Received by the editors April 7, 2006; accepted for publication (in revised form) by H. J. Woerdeman December 19, 2006; published electronically May 22, 2007. This work was supported by the Deutsche Forschungsgemeinschaft (DFG), the Heisenberg-Landau Program, and the Russian Foundation for Basic Research.
http://www.siam.org/journals/simax/29-2/65667.html

[†] Institut für Angewandte Mathematik, Universität Bonn, Wegelerstraße 6, D-53115 Bonn, Germany; SFB 611, Bonn; BiBoS, Bielefeld-Bonn; CERFIM, Locarno; Accademia di Architettura, USI, Mendrisio, Switzerland (albeverio@uni-bonn.de).

[‡] Bogoliubov Laboratory of Theoretical Physics, JINR, Joliot-Curie 6, 141980 Dubna, Moscow Region, Russia (motovilv@theor.jinr.ru).

[§] Open Technologies Inc., Obrucheva 30 Str. 1, 117997 Moscow, Russia (selin@ot.ru).

Moreover, if no assumptions are made about the location of $\sigma_0$ and $\sigma_1$ except the assumption (1.1), then condition (1.2) is sharp (see [6, Example 1.5]).

However, there are two important particular mutual positions of the spectral sets $\sigma_0$ and $\sigma_1$ that ensure the disjointness of the perturbed spectral sets $\sigma_0'$ and $\sigma_1'$ under conditions on $\|V\|$ much weaker than the general one of (1.2). The first of these two dispositions is the one where the sets $\sigma_0$ and $\sigma_1$ are subordinated, say

$$\sup \sigma_0 < \inf \sigma_1. \tag{1.3}$$

The second disposition corresponds to the case where one of the sets $\sigma_0$ and $\sigma_1$ is lying in a finite gap of the other set, say $\sigma_0$ lies in a finite gap of $\sigma_1$, which means that

$$\operatorname{conv}(\sigma_0) \cap \sigma_1 = \varnothing, \tag{1.4}$$

where $\operatorname{conv}(\sigma)$ denotes the convex hull of a set $\sigma \subset \mathbb{R}$. (We recall that by a finite gap of a closed Borel set $\Sigma$ on $\mathbb{R}$ one understands an open finite interval belonging to the complement $\mathbb{R} \setminus \Sigma$ of $\Sigma$ and such that both of its end points belong to $\Sigma$.)

It is known that if (1.3) holds, then for any bounded off-diagonal perturbation $V$ the interval $(\sup \sigma_0, \inf \sigma_1)$ belongs to the resolvent set of the perturbed operator $L = A + V$, and hence $\sigma_0' \subset (-\infty, \sup \sigma_0]$ and $\sigma_1' \subset [\inf \sigma_1, +\infty)$ (see [1, 4, 8]; cf. [5]). In the case of the disposition (1.4), it has been proven in [6] (see also [5]) that the gaps between $\sigma_0$ and $\sigma_1$ remain open if the off-diagonal perturbation $V$ satisfies the (sharp) condition

$$\|V\| < \sqrt{2}d.$$

Under this condition, the spectrum of $L = A + V$ consists of two disjoint components $\sigma_0'$ and $\sigma_1'$ such that

$$\sigma_0' \subset (\inf\, \sigma_0 - d, \sup\, \sigma_0 + d) \quad \text{and} \quad \sigma_1' \subset \mathbb{R} \setminus \Delta,$$

where $\Delta$ denotes the gap of $\sigma_1$ that contains $\sigma_0$. Notice that the norm bound $\|V\| < \sqrt{2}d$ is also sharp in the sense that, if it is violated, the spectrum of $L$ in the gap $\Delta$ may be empty at all (see [6, Example 1.6]).

Now assume that the perturbed spectral set $\sigma_0'$ contains an eigenvalue of the operator $L = A + V$ and let $f$, $f \neq 0$, be an eigenvector of $L$ corresponding to this eigenvalue. Denote by $\theta$ the (acute) angle between the vector $f$ and its projection $f_0 = \mathsf{E}_A(\sigma_0) f$ onto the spectral subspace $\mathfrak{H}_0 = \operatorname{Ran} \mathsf{E}_A(\sigma_0)$ of $A$ associated with the unperturbed spectral set $\sigma_0$.

Under the subordination condition (1.3), for any bounded off-diagonal perturbation $V$ the angle $\theta$ cannot exceed $\pi/4$. Moreover, the following sharp estimate holds:

$$\theta \leq \frac{1}{2} \arctan\left(\frac{2\|V\|}{d}\right) \quad \left(< \frac{\pi}{4}\right). \tag{1.5}$$

This bound is a simple corollary to the celebrated Davis–Kahan $\tan 2\Theta$ theorem [4] (see also [2, Theorem 5.1], [3, Theorem 6.1], and [7, Theorem 2.4]).

In the case of the spectral disposition (1.4), an a posteriori bound on the angle $\theta$ under condition $\|V\| < \sqrt{2}d$ follows from [6, Theorem 2.4]. This bound reads

$$\theta \leq \arctan\left(\frac{\|V\|}{\delta}\right), \tag{1.6}$$

where $\delta$ denotes the distance between the perturbed spectral set $\sigma_0'$ and unperturbed spectral set $\sigma_1$. Since $\delta$ may be arbitrarily small (see Example 2.5 below), the bound (1.6), in general, gives no a priori uniform estimate for $\theta$ except that $\theta < \pi/2$.

The present note is aimed just at giving an a priori sharp bound on the angle $\theta$ in the case of the disposition (1.4). In particular, we will prove that under condition $\|V\| < \sqrt{2}d$ this angle is strictly separated from $\pi/2$. Our main result is as follows.

Theorem 1.1. [illegible] $A$ [illegible] $\mathfrak{H}$ [illegible]

$$\operatorname{spec}(A) = \sigma_0 \cup \sigma_1, \quad \operatorname{dist}(\sigma_0, \sigma_1) = d > 0, \quad [illegible] \quad \operatorname{conv}(\sigma_0) \cap \sigma_1 = \varnothing.$$

[illegible] $V$ [illegible] $\mathfrak{H}$ [illegible] $\mathfrak{H} = \operatorname{Ran} \mathsf{E}_A(\sigma_0) \oplus \operatorname{Ran} \mathsf{E}_A(\sigma_1)$ [illegible]

$$\|V\| < \sqrt{2}d \tag{1.7}$$

[illegible] $L = A + V$ [illegible] $f$ [illegible]

$$z \in (\inf \sigma_0 - d, \sup \sigma_0 + d).$$

[illegible] $\theta$ [illegible] $f$ [illegible] $\mathsf{E}_A(\sigma_0)f$ [illegible] $\operatorname{Ran} \mathsf{E}_A(\sigma_0)$ [illegible]

$$\theta \le \arctan\left(\frac{\|V\|}{d}\right). \tag{1.8}$$

[illegible] 1.2. The bound (1.8) implies that under condition (1.7) the angle $\theta$ can never exceed the value of $\arctan\sqrt{2}$, i.e.,

$$\theta < \arctan\sqrt{2} \approx 0.304\,\pi.$$

We also remark that for $\|V\| < d$ the bound (1.8) follows from [9, Theorem 2].

Throughout the paper, by $\Xi(D, d, b)$ we will denote the function of three real variables $D$, $d$, and $b$ defined on the set

$$\Omega = \bigl\{(D, d, b) \mid \quad D > 0, \quad 0 < d \le D/2, \quad 0 \le b < \sqrt{dD}\bigr\}$$

by

$$\Xi(D, d, b) = \begin{cases} \tan^2\Bigl(\dfrac{1}{2}\arctan\dfrac{2b}{d}\Bigr) & \text{if } b^2 \le d\sqrt{D}\dfrac{\sqrt{D}-\sqrt{2d}}{2}, \\ 1 + \dfrac{2b^2}{D^2} - \dfrac{2}{D^2}\sqrt{(dD - b^2)\bigl((D-d)D - b^2\bigr)} & \\ \qquad\qquad\qquad \text{if } d\sqrt{D}\dfrac{\sqrt{D}-\sqrt{2d}}{2} < b^2 < dD. \end{cases} \tag{1.9}$$

Here and further on, by $\tan^2\theta$, $\theta \in \mathbb{R}$, we understand the square of the tangent of $\theta$, that is, $\tan^2\theta = (\tan\theta)^2$.

Theorem 1.1 appears to be a corollary to a more general statement (Theorem 3.2) that is proven under the condition

$$\|V\| < \sqrt{d|\Delta|}, \tag{1.10}$$

where $\Delta$ again denotes the (finite) gap of the set $\sigma_1$ that contains $\sigma_0$ and $|\Delta|$ stands for the length of the interval $\Delta$. It is known that if (1.10) holds, then the off-diagonal perturbation $V$ does not close the gaps between $\sigma_0$ and $\sigma_1$ (see [5, Theorem 1(i)]). Although condition (1.10) is in general weaker than (1.7), it involves the additional parameter $|\Delta|$. The claim of Theorem 3.2 is that under this condition the following inequality holds:

$$\tan\theta \le \bigl(\Xi(|\Delta|, d, \|V\|)\bigr)^{1/2}. \tag{1.11}$$

In particular, from formula (1.9) defining the function $\Xi$ one can see that if $|\Delta| > 2d$, then for $V$ small enough, namely for $V$ such that

$$\|V\|^2 \le d\sqrt{|\Delta|}\,\frac{\sqrt{|\Delta|}-\sqrt{2d}}{2},$$

the bound on $\theta$ is the same as the bound (1.5) prescribed by the $\tan 2\Theta$ theorem.

The paper is organized as follows. In section 2, we consider a three-dimensional version of the problem and prove the bound (1.11) in the case of $3\times 3$ matrices. The general finite- or infinite-dimensional case is studied in section 3. In the proof of the central result of this section, the one of Theorem 3.2, we essentially rely on Lemma 2.2 of section 2.

Throughout the paper, we use the standard notation $M^{\mathsf{T}}$ for the transpose of a matrix $M$.

**2. A three-dimensional case.** We start our consideration with the case where $\mathfrak{H} = \mathbb{C}^3$ and the operators $A$ and $V$ are $3\times 3$ matrices. Assume that

$$A = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \gamma_- & 0 \\ 0 & 0 & \gamma_+ \end{pmatrix} \quad\text{and}\quad V = \begin{pmatrix} 0 & b_- & b_+ \\ b_- & 0 & 0 \\ b_+ & 0 & 0 \end{pmatrix},$$

where

$$\lambda, \gamma_\pm, b_\pm \in \mathbb{R} \quad\text{and}\quad \gamma_+ > \gamma_-.$$

The matrices $A$ and $V$ are symmetric. Moreover, under the assumption that $\lambda \ne \gamma_\pm$, the matrix $V$ is off-diagonal with respect to the partition $\operatorname{spec}(A) = \sigma_0 \cup \sigma_1$ of the spectrum of $A$ into the disjoint sets

$$\sigma_0 = \{\lambda\} \quad\text{and}\quad \sigma_1 = \{\gamma_-, \gamma_+\}.$$

It is convenient for us to write the matrix $L = A + V$ in the following $2\times 2$ block form:

$$L = \begin{pmatrix} \lambda & B \\ B^* & A_1 \end{pmatrix}, \tag{2.1}$$

where $B$ and $A_1$ are $1\times 2$ and $2\times 2$ matrices given by

$$B = (\, b_- \quad b_+ \,), \quad A_1 = \begin{pmatrix} \gamma_- & 0 \\ 0 & \gamma_+ \end{pmatrix}, \tag{2.2}$$

respectively. Clearly, $\|V\| = \|B\| = \sqrt{|b_-|^2 + |b_+|^2}$.

Throughout this section, by $\Delta$ we will denote the spectral gap of the operator $A_1$ between its eigenvalues $\gamma_-$ and $\gamma_+$, i.e.,

$$\Delta = (\gamma_-, \gamma_+).$$

Lemma 2.1. [illegible] $L$ [illegible] (2.1) (2.2) [illegible] $\lambda \in \Delta$ [illegible]

$$\|B\| < \sqrt{d|\Delta|}, \tag{2.3}$$

[illegible] $|\Delta| = \gamma_+ - \gamma_-$ [illegible] $\Delta$ [illegible]

$$d = \operatorname{dist}(\sigma_0, \sigma_1) = \min\{\gamma_+ - \lambda, \lambda - \gamma_-\}.$$

[illegible] $L$ [illegible] $z$ [illegible] $\Delta$ [illegible]

$$\gamma_- < z_{\min} \le z \le z_{\max} < \gamma_+,$$

[illegible]

$$z_{\min} = \lambda - \|B\| \tan\left(\frac{1}{2} \arctan \frac{2\|B\|}{\gamma_+ - \lambda}\right), \tag{2.4}$$

$$z_{\max} = \lambda + \|B\| \tan\left(\frac{1}{2} \arctan \frac{2\|B\|}{\lambda - \gamma_-}\right). \tag{2.5}$$

[illegible]. Lemma 2.1 is an elementary corollary to [5, Theorem 3.2]. □

Lemma 2.2. [illegible] 2.1 [illegible] $z$ [illegible] $L$ [illegible] $\Delta$ [illegible] $f$ $f \neq 0$ [illegible] $Lf = zf$ [illegible] $\theta$ [illegible] $f$ [illegible] $f_0 = (1, 0, 0)^\mathsf{T}$ [illegible]

$$\tan^2 \theta \le \Xi(|\Delta|, d, \|B\|), \tag{2.6}$$

[illegible] $\Xi$ [illegible] (1.9)

[illegible]. Assume, without loss of generality, that $\gamma_+ = -\gamma_- = \gamma > 0$. Otherwise one can simply make the corresponding shift of the origin of the spectral parameter axis. Assume, in addition, that $B \neq 0$ and $\lambda \ge 0$. (There is no loss of generality in the latter assumption, since, for $\lambda < 0$, instead of $L$ one may consider the matrix $-L$.)

Thus, in the proof we will assume that

$$\Delta = (-\gamma, \gamma), \quad 0 \le \lambda < \gamma, \quad \text{and} \quad d = \gamma - \lambda.$$

Under the hypothesis that $\|B\| < \sqrt{d|\Delta|}$ $(= \sqrt{2d\gamma})$, from [5, Theorem 1(i)] it follows that if the eigenvalue $z$ of $L$ is in $\Delta$, then the corresponding eigenvector $f$, $Lf = zf$, may be chosen in the form

$$f = (1, x_-, x_+)^\mathsf{T},$$

with $x_\pm \in \mathbb{C}$ such that the matrix $X = (x_- \ x_+)^\mathsf{T}$ satisfies the Riccati equation

$$\lambda X - A_1 X + XBX = B^*. \tag{2.7}$$

Moreover,

$$z = \lambda + BX. \tag{2.8}$$

Taking into account (2.2), (2.7) and (2.8) imply

$$x_- = \frac{b_-}{\gamma + z} \quad \text{and} \quad x_+ = \frac{b_+}{-\gamma + z}. \tag{2.9}$$

Hence

$$\|X\|^2 = \frac{|b_-|^2}{(\gamma + z)^2} + \frac{|b_+|^2}{(-\gamma + z)^2}. \tag{2.10}$$

In addition, from (2.8) and (2.9) one concludes that $z$ is the solution to equation

$$z = \lambda + \frac{|b_-|^2}{\gamma + z} + \frac{|b_+|^2}{-\gamma + z}. \tag{2.11}$$

Let $t \in [0, 1]$ be such that

$$|b_+|^2 = t\|B\|^2 \tag{2.12}$$

and, hence,

$$|b_-|^2 = (1 - t)\|B\|^2. \tag{2.13}$$

Notice that under the assumptions we use, the points $z_{\min}$ of (2.4) and $z_{\max}$ of (2.5) can be written in the form

$$z_{\min} = \frac{\gamma + \lambda}{2} - \sqrt{\frac{(\gamma - \lambda)^2}{4} + \|B\|^2}, \tag{2.14}$$

$$z_{\max} = -\frac{\gamma - \lambda}{2} + \sqrt{\frac{(\gamma + \lambda)^2}{4} + \|B\|^2}. \tag{2.15}$$

It is easy to see that, given the value of $\|B\|$, for $t$ in (2.12) and (2.13) varying between 0 and 1 the solution $z$ to (2.11) fills the whole interval $[z_{\min}, z_{\max}]$. Moreover, with $t$ decreasing from 1 to 0 the value of $z$ is continuously and monotonously increasing from $z_{\min}$ to $z_{\max}$.

On the other hand, one can express $t$ through $z$. With $|b_\pm|$ given by (2.12) and (2.13), from (2.11) it follows that

$$t = \frac{1}{2\gamma\|B\|^2}[(z - \lambda)(z^2 - \gamma^2) - \|B\|^2(z - \gamma)]. \tag{2.16}$$

Taking this into account, we rewrite expression (2.10) in the form

$$\|X\|^2 = \varphi(z), \tag{2.17}$$

where the function $\varphi$ is given by

$$\varphi(z) = \frac{\|B\|^2 + 2(\lambda - z)z}{\gamma^2 - z^2}. \tag{2.18}$$

That is, given the value of $\|B\|$, the norm of the solution $X$ to the Riccati equation (2.7) may be considered as a function of the only variable $z$ that runs through the interval $[z_{\min}, z_{\max}]$.

There is a single point $z_0$ within the interval $(-\gamma, \gamma)$ where the derivative of the function $\varphi(z)$ is zero, namely

$$z_0 = \begin{cases} 0 & \text{if} \quad \lambda = 0, \\ \dfrac{2\gamma^2 - \|B\|^2}{2\lambda} - \sqrt{\left(\dfrac{2\gamma^2 - \|B\|^2}{2\lambda}\right)^2 - \gamma^2} & \text{if} \quad \lambda > 0. \end{cases} \tag{2.19}$$

It provides the function $\varphi(z)$ with a maximum.

One concludes by inspection that inequality (2.3) (along with the assumptions $\lambda \geq 0$ and $B \neq 0$) implies

$$z_0 < z_{\max}.$$

At the same time, $z_0 \leq z_{\min}$ if $0 < \|B\| \leq \beta$ and $z_0 > z_{\min}$ if $\beta < \|B\| < \sqrt{2d\gamma}$, where

$$\beta = \left[(\gamma - \lambda)\sqrt{\gamma}(\sqrt{\gamma} - \sqrt{\gamma - \lambda})\right]^{1/2} = \left[d\sqrt{|\Delta|}\frac{\sqrt{|\Delta|} - \sqrt{2d}}{2}\right]^{1/2}. \tag{2.20}$$

Therefore,

$$\max_{z \in [z_{\min}, z_{\max}]} \varphi(z) = \varphi(z_{\min}) \quad \text{if} \quad 0 < \|B\| \leq \beta \tag{2.21}$$

and

$$\max_{z \in [z_{\min}, z_{\max}]} \varphi(z) = \varphi(z_0) \quad \text{if} \quad \beta < \|B\| < \sqrt{d|\Delta|}. \tag{2.22}$$

By substituting (2.14) and (2.19) into (2.18), one arrives at

$$\varphi(z_{\min}) = \frac{d^2}{2\|B\|^2}\left(1 + \frac{2\|B\|^2}{d^2} - \sqrt{1 + \frac{4\|B\|^2}{d^2}}\right) = \tan^2\left(\frac{1}{2}\arctan\frac{2\|B\|}{d}\right) \tag{2.23}$$

and

$$\varphi(z_0) = 1 + \frac{2\|B\|^2}{|\Delta|^2} - \frac{2}{|\Delta|^2}\sqrt{(d|\Delta| - \|B\|^2)\big((|\Delta| - d)|\Delta| - \|B\|^2\big)}, \tag{2.24}$$

respectively. To get (2.6), it remains only to observe that $\tan\theta = \|X\|$, which by combining (2.17) and (2.21)–(2.24) means

$$\tan\theta \leq \left(\max_{z \in [z_{\min}, z_{\max}]} \varphi(z)\right)^{1/2} = \Xi(|\Delta|, d, \|B\|)^{1/2}.$$

The proof is complete. □

[illegible] 2.3. The bound (2.6) is optimal in the sense that given the values of $|\Delta| > 0$, $d \in (0, |\Delta|/2)$, and $\|B\| < \sqrt{d|\Delta|}$, it is possible to choose a matrix $L$ of the form (2.1), (2.2) such that for the eigenvector $f = (1, x_-, x_+)^\mathsf{T}$ associated with the (only) eigenvalue $z$ of $L$ within the interval $(\gamma_-, \gamma_+)$ inequality (2.6) turns into equality.

To prove this statement, set $\gamma = \frac{|\Delta|}{2}$, $\gamma_\pm = \pm\gamma$, and $\lambda = \gamma - d$. If $\|B\| \le \beta$, where $\beta$ is given by (2.20), then choose $b_- = 0$ and $b_+ = \|B\|$. Observe that in this case $z = z_{\min}$, and hence by (2.21) such a choice of $b_\pm$ just provides $\|X\|^2 = x_-^2 + x_+^2$ with its maximal possible value; i.e., the equalities $\tan^2\theta = \varphi(z_{\min}) = \Xi(|\Delta|, d, \|B\|)$ hold. If $\|B\| > \beta$, first compute $t$ by formula (2.16) for $z = z_0$ with $z_0$ given by (2.19). Then introduce $b_+ = \sqrt{t}\|B\|$ and $b_- = \sqrt{1-t}\,\|B\|$. In such a case, $z = z_0$ is the eigenvalue of the matrix $L$ in $\Delta$, and we have the equality $\tan^2\theta = \varphi(z_0)$; that is, again the equality $\tan^2\theta = \Xi(|\Delta|, d, \|B\|)$ holds.

Example 2.4. Again assume that $\gamma_+ = -\gamma_- = \frac{|\Delta|}{2} > 0$. Assume, in addition, that $\lambda = 0$ and $b_+ = b_- = \frac{b}{\sqrt{2}}$ for some $b \ge 0$. From (2.11), it is easy to see that in this case $z = 0$ is the (only) eigenvalue of the matrix $L$ within the interval $\Delta$. Moreover, for the corresponding eigenvector $f = (1, x_-, x_+)^\mathsf{T}$, by (2.9) one infers that $x_- = -\frac{b}{\sqrt{2}d}$ and $x_+ = \frac{b}{\sqrt{2}d}$, taking into account that $\gamma_- = -d$ and $\gamma_+ = d$. Since $\|B\| = b$, the equality $\tan\theta = \sqrt{|x_-|^2 + |x_+|^2}$ yields

$$\tan\theta = \frac{\|B\|}{d}.$$

Notice that in this example $\Xi(|\Delta|, d, \|B\|) = \Xi(2d, d, \|B\|) = \frac{\|B\|^2}{d^2}$, and thus the equality $\tan^2\theta = \Xi(|\Delta|, d, \|B\|)$ holds, too.

Example 2.5. Consider a matrix $L$ of the form (2.1) with $\gamma_-$, $\gamma_+$, and $\lambda$ as in Example 2.4, that is, with $\gamma_+ = -\gamma_- = d > 0$ and $\lambda = 0$. Set $b_+ = 0$ and let $b_-$ satisfy the inequalities $0 \le b_- < \sqrt{d|\Delta|}$. Obviously, $\|V\| = b_-$, $|\Delta| = 2d$, and thus we have $\|V\| < \sqrt{2}d$. The eigenvalue $z$ of the matrix $L$ in the interval $\Delta$ (which is the corresponding solution to (2.11)) simply coincides with $z_{\max}$ (cf. formula (2.15)),

$$z = -\frac{d}{2} + \sqrt{\frac{d^2}{4} + \|V\|^2}.$$

Clearly, $z \to d$ as $\|V\| \to \sqrt{2}d$. That is, in this case the distance $\delta = \operatorname{dist}(\sigma_0', \sigma_1)$ between the perturbed spectral set $\sigma_0' = \{z\}$ and unperturbed spectral set $\sigma_1 = \{-d, d\}$ can be made arbitrarily small.

**3. General case.** Recall that by a finite spectral gap of a self-adjoint operator $T$ one understands an *open* finite interval on $\mathbb{R}$ lying in the resolvent set of $T$ and being such that both of its end points belong to the spectrum of $T$.

In what follows, we adopt the following hypothesis.

Hypothesis 3.1. Let the Hilbert space $\mathfrak{H}$ be decomposed into the orthogonal sum of two subspaces, i.e.,

$$\mathfrak{H} = \mathfrak{H}_0 \oplus \mathfrak{H}_1. \tag{3.1}$$

Assume that, with respect to the decomposition (3.1), a self-adjoint operator $L$ on $\mathfrak{H}$ reads as a $2\times 2$ operator block matrix,

$$L = \begin{pmatrix} A_0 & B \\ B^* & A_1 \end{pmatrix}, \quad \operatorname{Dom}(L) = \mathfrak{H}_0 \oplus \operatorname{Dom}(A_1),$$

where $A_0$ is a bounded self-adjoint operator on $\mathfrak{H}_0$, $A_1$ a possibly unbounded self-adjoint operator on $\mathfrak{H}_1$, and $B$ a bounded operator from $\mathfrak{H}_1$ to $\mathfrak{H}_0$. Assume, in addition, that $A_1$ has a finite spectral gap $\Delta = (\gamma_-, \gamma_+)$, $\gamma_- < \gamma_+$, the spectrum of $A_0$ lies in $\Delta$, i.e., $\operatorname{spec}(A_0) \subset \Delta$, and

$$\|B\| < \sqrt{d|\Delta|}, \tag{3.2}$$

where

$$d = \operatorname{dist}(\operatorname{spec}(A_0), \operatorname{spec}(A_1)).$$

If $f$ is a nonzero element of the Hilbert space $\mathfrak{H}$ and $\mathfrak{K}$ is a subspace of $\mathfrak{H}$, by the angle between $f$ and $\mathfrak{K}$ we understand the acute angle $\theta$ between $f$ and its orthogonal projection $f_{\mathfrak{K}}$ onto $\mathfrak{K}$, that is, $\theta = \arccos(\|f_{\mathfrak{K}}\|/\|f\|)$.

THEOREM 3.2. [illegible] 3.1 [illegible] $L$ [illegible] $\Delta$ [illegible] $f$ [illegible] $L$ [illegible] $\theta$ [illegible] $f$ [illegible] $\mathfrak{H}_0$ [illegible]

$$\tan^2\theta \le \Xi(|\Delta|, d, \|B\|), \tag{3.3}$$

[illegible] $\Xi$ [illegible] (1.9)

[illegible]. Assume that the eigenvector $f = f_0 \oplus f_1$, $f_0 \in \mathfrak{H}_0$, $f_1 \in \operatorname{Dom}(A_1)$, of the operator $L$ is associated with an eigenvalue $z \in \Delta$. Then the following equalities hold:

$$A_0 f_0 + B f_1 = z f_0, \tag{3.4}$$

$$B^* f_0 + A_1 f_1 = z f_1. \tag{3.5}$$

Taking into account that $z$ is in the resolvent set of $A_1$, from (3.5) it follows that

$$f_1 = -(A_1 - z)^{-1} B^* f_0. \tag{3.6}$$

Hence, $f_0 \neq 0$ (otherwise, for $f_0 = 0$, one would have $f_1 = 0$ and then $f = 0$). Equations (3.4) and (3.6) yield

$$A_0 f_0 - B(A_1 - z)^{-1} B^* f_0 = z f_0,$$

which implies

$$\langle A_0 f_0, f_0\rangle - \langle B(A_1 - z)^{-1} B^* f_0, f_0\rangle = z\|f_0\|^2. \tag{3.7}$$

From now on, suppose that

$$\|f_0\| = 1 \tag{3.8}$$

and set $\lambda = \langle A_0 f_0, f_0\rangle$. Clearly,

$$\lambda \in [\inf \operatorname{spec}(A_0), \sup \operatorname{spec}(A_0)]. \tag{3.9}$$

By the spectral theorem, we have

$$\langle B(A_1 - z)^{-1} B^* f_0, f_0\rangle = \int_{\mathbb{R}\setminus(\gamma_-,\gamma_+)} \frac{\langle d\mathsf{E}_{A_1}(\mu) B^* f_0, B^* f_0\rangle}{\mu - z}, \tag{3.10}$$

where $\mathsf{E}_{A_1}(\mu)$, $\mu \in \mathbb{R}$, denotes the spectral family of $A_1$. Let

$$\Delta_- = (-\infty, \gamma_-] \quad \text{and} \quad \Delta_+ = [\gamma_+, \infty).$$

By the mean value theorem, there are real numbers $\mu_- \le \gamma_-$ and $\mu_+ \ge \gamma_+$ such that

$$\int_{\Delta_\pm} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{\mu - z} = \frac{\langle \mathsf{E}_{A_1}(\Delta_\pm)B^*f_0, B^*f_0\rangle}{\mu_\pm - z} = \frac{\|\mathsf{E}_{A_1}(\Delta_\pm)B^*f_0\|^2}{\mu_\pm - z}, \tag{3.11}$$

respectively. Introduce the nonnegative numbers $b_\pm$ by

$$b_\pm = \sqrt{\alpha_\pm}\|\mathsf{E}_{A_1}(\Delta_\pm)B^*f_0\|, \tag{3.12}$$

where

$$\alpha_\pm = \frac{|\gamma_\pm - z|}{|\mu_\pm - z|} \le 1. \tag{3.13}$$

Obviously,

$$\int_{\Delta_\pm} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{\mu - z} = \frac{b_\pm^2}{\gamma_\pm - z}. \tag{3.14}$$

Thus, taking into account (3.9), (3.10), and (3.11), (3.7) turns into

$$\lambda - \frac{b_-^2}{\gamma_- - z} - \frac{b_+^2}{\gamma_+ - z} = 0. \tag{3.15}$$

At the same time, by (3.6) we have

$$\|f_1\|^2 = \int_{\mathbb{R}\setminus(\gamma_-,\gamma_+)} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{(\mu - z)^2}. \tag{3.16}$$

The contributions of the intervals $(-\infty, \gamma_-]$ and $[\gamma_+, \infty)$ to the integral on the right-hand side of (3.16) are estimated separately. For the first interval, one derives

$$\int_{\Delta_-} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{(\mu - z)^2} \le \frac{1}{z - \gamma_-}\int_{\Delta_-} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{z - \mu},$$

which by (3.14) means

$$\int_{\Delta_-} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{(\mu - z)^2} \le \frac{b_-^2}{(\gamma_- - z)^2}. \tag{3.17}$$

In a similar way, one concludes that

$$\int_{\Delta_+} \frac{\langle d\mathsf{E}_{A_1}(\mu)B^*f_0, B^*f_0\rangle}{(\mu - z)^2} \le \frac{b_+^2}{(\gamma_+ - z)^2}. \tag{3.18}$$

Then by combining (3.16), (3.17), and (3.18) one infers that

$$\|f_1\|^2 \le x_-^2 + x_+^2, \tag{3.19}$$

where

$$x_\pm = -\frac{b_\pm}{\gamma_\pm - z}. \tag{3.20}$$

From (3.15), (3.20), it follows that the vector $y = (1, x_-, x_+)^{\mathsf{T}}$ is an eigenvector of the $3 \times 3$ matrix

$$\widetilde{L} = \begin{pmatrix} \lambda & b_- & b_+ \\ b_- & \gamma_- & 0 \\ b_+ & 0 & \gamma_+ \end{pmatrix}$$

associated with the eigenvalue $z$, that is, $\widetilde{L}y = zy$. By (3.9), for $\delta = \operatorname{dist}(\lambda, \{\gamma_-, \gamma_+\})$ we have

$$d \le \delta \le \frac{|\Delta|}{2}. \tag{3.21}$$

In addition, by (3.12) the square of the norm $\|\widetilde{B}\| = \sqrt{b_-^2 + b_+^2}$ of the $1 \times 2$ matrix-row $\widetilde{B} = (b_- \quad b_+)$ reads

$$\|\widetilde{B}\|^2 = \alpha_-^2 \langle \mathsf{E}_{A_1}(\Delta_-) B^* f_0, B^* f_0 \rangle + \alpha_+^2 \langle \mathsf{E}_{A_1}(\Delta_+) B^* f_0, B^* f_0 \rangle,$$

and hence

$$\begin{aligned} \|\widetilde{B}\|^2 &\le \langle \mathsf{E}_{A_1}(\Delta_-) B^* f_0, B^* f_0 \rangle + \langle \mathsf{E}_{A_1}(\Delta_+) B^* f_0, B^* f_0 \rangle \\ &= \langle B^* f_0, B^* f_0 \rangle = \|B^* f_0\|^2 \\ &\le \|B\|^2, \end{aligned} \tag{3.22}$$

taking into account first (3.13) and then (3.8). By the hypothesis, inequality (3.2) holds. Combining (3.21) and (3.22) with (3.2) implies

$$\|\widetilde{B}\|^2 < \sqrt{\delta |\Delta|}. \tag{3.23}$$

By Lemma 2.2, one then concludes that $x_-^2 + x_+^2 \le \Xi(|\Delta|, \delta, \|\widetilde{B}\|)$, which by (3.8) and (3.19) implies that

$$\tan^2 \theta \le \Xi(|\Delta|, \delta, \|\widetilde{B}\|). \tag{3.24}$$

Given $|\Delta| > 0$, $d \in (0, |\Delta|/2]$, and $\|B\|$ satisfying (3.2), it is easy to see that the function $\Xi(|\Delta|, \delta, \|\widetilde{B}\|)$ is monotonously increasing with increasing $\|\widetilde{B}\|$, $\|\widetilde{B}\| \le \|B\|$. For $d < |\Delta|/2$, it also monotonously increases if $\delta$ decreases from $\frac{|\Delta|}{2}$ to $d$. Therefore, from (3.24) it follows that $\tan^2 \theta \le \Xi(|\Delta|, d, \|B\|)$, completing the proof. □

*Remark* 3.3. The bound (3.3) is optimal. This follows from Remark 2.3.

*Remark* 3.4. Notice that under condition $\|B\| < \sqrt{d(|\Delta| - d)}$ from [9, Theorem 5.3] the operator angle $\Theta$ between the unperturbed and perturbed spectral subspaces $\operatorname{Ran} \mathsf{E}_A(\sigma_0)$ and $\operatorname{Ran} \mathsf{E}_L(\sigma_0')$ satisfies the following (sharp) estimate:

$$\Theta \le \frac{1}{2} \arctan \kappa(\|B\|), \tag{3.25}$$

where the function $\kappa(b)$ is defined for $0 \le b < \sqrt{d(|\Delta| - d)}$ by

$$\kappa(b) = \begin{cases} \dfrac{2b}{d} & \text{if } b \le \sqrt{\dfrac{d}{2}\left(\dfrac{|\Delta|}{2} - d\right)}, \\[2ex] \dfrac{b\dfrac{|\Delta|}{2} + \sqrt{d(|\Delta| - d)\left[\left(\dfrac{|\Delta|}{2} - d\right)^2 + b^2\right]}}{d(|\Delta| - d) - b^2} & \text{if } b > \sqrt{\dfrac{d}{2}\left(\dfrac{|\Delta|}{2} - d\right)}. \end{cases}$$

Surely, the bound (3.25) implies the corresponding estimate for the angle $\theta$:

$$\theta \leq \frac{1}{2} \arctan \kappa(\|B\|) \quad \text{whenever} \quad \|B\| < \sqrt{d(|\Delta| - d)}. \tag{3.26}$$

One observes by inspection that

$$\Xi(|\Delta|, d, b) \leq \tan^2 \left( \frac{1}{2} \arctan \kappa(b) \right), \quad 0 \leq b < \sqrt{d(|\Delta| - d)}.$$

Moreover, if $|\Delta| > 2d$, then for $\sqrt{\frac{d}{2}(\frac{|\Delta|}{2} - d)} < b < \sqrt{d(|\Delta| - d)}$ the strict inequality $\Xi(|\Delta|, d, b) < \tan^2 \left(\frac{1}{2} \arctan \kappa(b)\right)$ holds. Therefore, the bound (3.26) is not optimal in the case of eigenvectors.

Now we are in position to prove Theorem 1.1. This theorem appears to be a simple corollary to Theorem 3.2.

[illegible] 1.1. Set $\mathfrak{H}_0 = \operatorname{Ran} \mathsf{E}_A(\sigma_0)$ and $\mathfrak{H}_0 = \operatorname{Ran} \mathsf{E}_A(\sigma_1)$. With respect to the orthogonal decomposition $\mathfrak{H} = \mathfrak{H}_0 \oplus \mathfrak{H}_1$, the operators $A$ and $V$ read as $2 \times 2$ block operator matrices,

$$A = \begin{pmatrix} A_0 & 0 \\ 0 & A_1 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 0 & B \\ B^* & 0 \end{pmatrix},$$

where $B = V|_{\mathfrak{H}_1}$; $\operatorname{Dom}(A) = \mathfrak{H}_0 \oplus \operatorname{Dom}(A_1)$ and $\operatorname{Dom}(L) = \operatorname{Dom}(A)$. Assume that $\Delta$ is a gap of the set $\sigma_1$ that contains the whole set $\sigma_0$. Surely, the length $|\Delta|$ of this gap satisfies the estimate $|\Delta| \geq 2d$, and the bound (1.7) implies the inequality $\|B\| < \sqrt{d|\Delta|}$. Then by Theorem 3.2 we have

$$\tan^2 \theta \leq \Xi(|\Delta|, d, \|V\|),$$

taking into account that $\|V\| = \|B\|$. Now it remains only to observe that $\Xi(D, d, \|V\|)$ is a nonincreasing function of the variable $D$, $D \geq 2d$. For $D$ varying in the interval $[2d, \infty)$, it achieves its maximal value just at $D = 2d$, and this value equals

$$\max_{D\colon D \geq 2d} \Xi(|\Delta|, d, \|V\|) = \frac{\|V\|^2}{d^2}.$$

Thus, the following inequality holds:

$$\tan \theta \leq \frac{\|V\|}{d}.$$

The proof is complete. □

[illegible] 3.5. Example 2.4 shows that the bound (1.8) is sharp.

## REFERENCES

[1] V. Adamyan, H. Langer, and C. Tretter, *Existence and uniqueness of contractive solutions of some Riccati equations*, J. Funct. Anal., 179 (2001), pp. 448–473.

[2] C. Davis, *The rotation of eigenvectors by a perturbation*, J. Math. Anal. Appl., 6 (1963), pp. 159–173.

[3] C. Davis, *The rotation of eigenvectors by a perturbation.* II, J. Math. Anal. Appl., 11 (1965), pp. 20–27.

[4] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation.* III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

[5] V. Kostrykin, K. A. Makarov, and A. K. Motovilov, *On the existence of solutions to the operator Riccati equation and the tan Θ theorem*, Integral Equations Operator Theory, 51 (2005), pp. 121–140.
[6] V. Kostrykin, K. A. Makarov, and A. K. Motovilov, *Perturbation of spectra and spectral subspaces*, Trans. Amer. Math. Soc., 359 (2007), pp. 77–89.
[7] V. Kostrykin, K. A. Makarov, and A. K. Motovilov, *A generalization of the* tan 2Θ *theorem*, Oper. Theory Adv. Appl., 149 (2004), pp. 349–372.
[8] H. Langer and C. Tretter, *Diagonalization of certain block operator matrices and applications to Dirac operators*, Oper. Theory Adv. Appl., 122 (2001), pp. 331–358.
[9] A. K. Motovilov and A. V. Selin, *Some sharp norm estimates in the subspace perturbation problem*, Integral Equations Operator Theory, 56 (2006), pp. 511–542.
[10] F. Riesz and B. Sz.-Nagy, *Leçons d'analyse fonctionnelle*, 2nd ed., Académiai Kiado, Budapest, 1953.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 699–711

# ON PSEUDOSPECTRA AND POWER GROWTH*

THOMAS RANSFORD†

**Abstract.** The celebrated Kreiss matrix theorem is one of several results relating the norms of the powers of a matrix to its pseudospectra (i.e., the level curves of the norm of the resolvent). But to what extent do the pseudospectra actually *determine* the norms of the powers? Specifically, let $A, B$ be square matrices such that, with respect to the usual operator norm $\|\cdot\|$, we have $\|(zI-A)^{-1}\| = \|(zI-B)^{-1}\|$ $(z \in \mathbb{C})$. (Call this $(*)$.) Then it is known that $1/2 \leq \|A\|/\|B\| \leq 2$. Are there similar bounds for $\|A^n\|/\|B^n\|$ for $n \geq 2$? Does the answer change if $A, B$ are diagonalizable? What if $(*)$ holds, not just for the norm $\|\cdot\|$, but also for higher-order singular values? What if we use norms other than the usual operator norm? The answers to all these questions turn out to be negative, and in a rather strong sense.

**Key words.** matrix, norm, spectral radius, eigenvalue, singular value, pseudospectra

**AMS subject classifications.** Primary 47A10; Secondary 15A18, 15A60, 65F15

**DOI.** 10.1137/060658126

## 1. Introduction and statement of results.

Let $N \geq 1$, let $\mathbb{C}^N$ be complex Euclidean $N$-space, and let $\mathbb{C}^{N\times N}$ be the algebra of complex $N \times N$ matrices. We write $|\cdot|$ for the Euclidean norm on $\mathbb{C}^N$, defined by $|x| := (\sum_1^N |x_j|^2)^{1/2}$, and write $\|\cdot\|$ for the associated operator norm on $\mathbb{C}^{N\times N}$, defined by $\|A\| := \sup\{|Ax| : |x| = 1\}$.

It is well known that, given $A \in \mathbb{C}^{N\times N}$, the long-term growth of the norms of powers of $A$ is governed by the spectral radius $\rho(A)$. Indeed, by the spectral radius formula, we have

$$\|A^n\| \geq \rho(A)^n \quad (n \geq 1) \qquad \text{and} \qquad \lim_{n\to\infty} \|A^n\|^{1/n} = \rho(A).$$

However, in the shorter term, $\|A^n\|$ may well be significantly larger than $\rho(A)^n$. The recent book of Trefethen and Embree [4] contains an account of these transient effects, illustrated by examples drawn from many different fields. One of the main themes of the book is that, to accurately predict the growth of $\|A^n\|$, it is important to study not only the spectrum of $A$, but also its pseudospectra, which we now define.

Given $A \in \mathbb{C}^{N\times N}$ and $\epsilon > 0$, the $\epsilon$ [illegible] of $A$ is defined to be the set

$$\sigma_\epsilon(A) := \{z \in \mathbb{C} : \|(zI - A)^{-1}\| > \epsilon^{-1}\}.$$

Here and in what follows we adopt the useful convention that $\|(zI - A)^{-1}\| = \infty$ if $z \in \sigma(A)$, the spectrum of $A$. Thus $\sigma_\epsilon(A)$ shrinks to $\sigma(A)$ as $\epsilon \downarrow 0$. From a knowledge of the pseudospectra of $A$, it is possible to deduce both upper and lower bounds on the growth of $\|A^n\|$. A well-known result of this type is the Kreiss matrix theorem. We refer to [4] for this and several other such results. In addition, there are efficient methods for numerical computation of pseudospectra (see [4, Chapter IX]), so this approach is highly practical.

---

*Received by the editors April 25, 2006; accepted for publication (in revised form) by J. H. Brandts January 2, 2007; published electronically June 5, 2007. This research was supported by grants from NSERC and the Canada Research Chairs program.

http://www.siam.org/journals/simax/29-3/65812.html

†Département de Mathématiques et de Statistique, Université Laval, Québec City, G1K 7P4, QC, Canada (ransford@mat.ulaval.ca).

The purpose of this paper is to show that, in predicting power growth, not even pseudospectra tell the whole story. The issue was already addressed by Greenbaum and Trefethen in [3] (see also [4, section 47]). Suppose that two $N \times N$ matrices $A, B$ have identical pseudospectra, i.e., that

$$\|(zI - A)^{-1}\| = \|(zI - B)^{-1}\| \qquad (z \in \mathbb{C}). \tag{1.1}$$

Does it then follow that $\|p(A)\| = \|p(B)\|$ for every polynomial $p$? In particular, can we conclude that $\|A^n\| = \|B^n\|$ for all $n \geq 1$? The answer is no. An example was given in [3] (and again in [4]) of two matrices $A, B$ with identical pseudospectra such that $\|A\| = 1$ and $\|B\| = \sqrt{2}$. But this example leaves the following basic questions unresolved.

**1.1. What about higher powers?** By adapting the Greenbaum–Trefethen example, one can construct, for each $\epsilon > 0$, matrices $A, B$ with identical pseudospectra such that $\|A\|/\|B\| > 2 - \epsilon$. On the other hand, it is known that if $A, B$ satisfy (1.1), then we must have

$$1/2 \leq \|A\|/\|B\| \leq 2. \tag{1.2}$$

(For a proof, see [4, pp. 168–169]; see also the remark after Theorem 5.1 below.) Are there similar bounds for $\|A^n\|/\|B^n\|$ for $n \geq 2$? If this were the case, then we could justifiably say that pseudospectra determine power norms, at least up to a constant factor. However, our first result answers this question negatively, and in a fairly strong sense.

Recall that a (finite or infinite) sequence $(\alpha_k)$ is called [illegible] if $\alpha_{k+l} \leq \alpha_k \alpha_l$ for all $k, l$ for which the inequality makes sense. For example, the sequence $(\|A^k\|)_{k \geq 1}$ is submultiplicative for every matrix $A$.

THEOREM 1.1. [illegible] $n \geq 2$ [illegible] $\alpha_2, \dots, \alpha_n$ [illegible] $\beta_2, \dots, \beta_n$ [illegible] $N \geq 1$ [illegible] $A, B \in \mathbb{C}^{N \times N}$ [illegible]

$$\|(zI - A)^{-1}\| = \|(zI - B)^{-1}\| \qquad (z \in \mathbb{C}) \tag{1.3}$$

[illegible]

$$\|A^k\| = \alpha_k \quad \text{[illegible]} \quad \|B^k\| = \beta_k \qquad (k = 2, \dots, n). \tag{1.4}$$

[illegible] $N = 2n + 3$

This shows that matrices can have identical pseudospectra and yet their second and higher powers have norms that are completely unrelated to each other.

**1.2. What about diagonalizable matrices?** The matrices $A, B$ in the Greenbaum–Trefethen example are nilpotent, as are those constructed in the proof of Theorem 1.1 above. Obviously, these are rather special. What happens if, instead, we consider more generic matrices, for example, diagonalizable matrices? (By "diagonalizable" we mean similar to a diagonal matrix.) Could it be that, for such matrices at least, the pseudospectra completely determine the power growth? Until now, no counterexample was known. We obtain one by combining the construction in Theorem 1.1 with a perturbation argument.

THEOREM 1.2. [illegible] $n \geq 2$ [illegible] $\alpha_2, \dots, \alpha_n$ [illegible] $\beta_2, \dots, \beta_n$ [illegible] $\epsilon > 0$ [illegible] $N \geq 1$ [illegible] diagonalizable [illegible] $A, B \in \mathbb{C}^{N \times N}$ [illegible]

$$\|(zI - A)^{-1}\| = \|(zI - B)^{-1}\| \qquad (z \in \mathbb{C}) \tag{1.5}$$

$$\alpha_k - \epsilon < \|A^k\| < \alpha_k + \epsilon \qquad \beta_k - \epsilon < \|B^k\| < \beta_k + \epsilon \qquad (k = 2, \ldots, n). \tag{1.6}$$

**1.3. What if we use "higher-order" pseudospectra?** Given a matrix $A \in \mathbb{C}^{N\times N}$, its $s_1(A), \ldots, s_N(A)$ are the square roots of the eigenvalues of $A^*A$, listed in decreasing order. In particular, $s_1(A) = \|A\|$. One of the principal methods for computing the pseudospectra of $A$ is to calculate the singular values of $zI - A$ for $z \in \mathbb{C}$ (once done for one value of $z$, it is relatively inexpensive to do for many $z$), and then use the fact that

$$\|(zI - A)^{-1}\| = s_1\Big((zI - A)^{-1}\Big) = \frac{1}{s_N(zI - A)}.$$

It is thus reasonable to ask whether, by retaining the other singular values of the resolvent $(zI - A)^{-1}$, it is possible to determine $\|A^n\|$ for values of $n \geq 2$. The following result gives a partial positive answer.

THEOREM 1.3. $N \geq 1$ $A, B \in \mathbb{C}^{N\times N}$

$$s_j\Big((zI - A)^{-1}\Big) = s_j\Big((zI - B)^{-1}\Big) \qquad (z \in \mathbb{C},\ j = 1, \ldots, N). \tag{1.7}$$

$p$

$$\frac{1}{\sqrt{N}} \leq \frac{\|p(A)\|}{\|p(B)\|} \leq \sqrt{N}. \tag{1.8}$$

It would be interesting to know if these bounds can be improved so as to be independent of $N$. However, even if this were the case, the theorem would be a bit unrealistic, because it would require us to keep track of all $N$ singular values, which is probably too expensive in practice. Is there an analogous result where, by keeping track of just a few singular values, we can obtain inequalities like (1.8) at least for polynomials of low degree? The following generalization of Theorem 1.1 gives a negative answer.

THEOREM 1.4. $n \geq 2$ $\alpha_2, \ldots, \alpha_n$ $\beta_2, \ldots, \beta_n$ $m \geq 1$ $N \geq 1$ $A, B \in \mathbb{C}^{N\times N}$

$$s_j\Big((zI - A)^{-1}\Big) = s_j\Big((zI - B)^{-1}\Big) \qquad (z \in \mathbb{C},\ j = 1, \ldots, m) \tag{1.9}$$

$$\|A^k\| = \alpha_k \qquad \|B^k\| = \beta_k \qquad (k = 2, \ldots, n). \tag{1.10}$$

$N = (m + 1)(n + 2) - 1$

**1.4. What about other norms?** Though the Euclidean-norm case is undoubtedly the most important one, there are instances where it is more appropriate to consider pseudospectra defined with respect to other norms. In [4, sections 56, 57], several examples are given based on the 1-norm on $\mathbb{C}^N$, defined by $|x|_1 := \sum_{j=1}^N |x_j|$ and the associated operator norm $\|\cdot\|_1$, given by $\|A\|_1 := \sup\{|Ax|_1 : |x|_1 = 1\}$. There is no analogue of the Greenbaum–Trefethen example for this norm, because of the following theorem.

THEOREM 1.5. [illegible] $N \geq 1$ [illegible] $A, B \in \mathbb{C}^{N\times N}$ [illegible]

$$\|(zI - A)^{-1}\|_1 = \|(zI - B)^{-1}\|_1 \qquad (z \in \mathbb{C}). \tag{1.11}$$

[illegible] $\|A\|_1 = \|B\|_1$

Can we also deduce that $\|A^n\|_1 = \|B^n\|_1$ for $n \geq 2$? Once again, the answer turns out to be no, and not just for $\|\cdot\|_1$, but for a whole variety of possible norms. To make this precise, it is convenient to introduce some notation and terminology.

Given square matrices $A, B$, perhaps of different sizes, we shall write $A \oplus B$ for the block matrix

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}.$$

A norm $|||\cdot|||$ on $\mathbb{C}^{N\times N}$ will be called [illegible] if it satisfies the following three conditions:

- $|||\cdot|||$ is an algebra norm, i.e., $|||AB||| \leq |||A|||.|||B|||$ for all $A, B \in \mathbb{C}^{N\times N}$ and $|||I||| = 1$;
- every permutation matrix $Q \in \mathbb{C}^{N\times N}$ satisfies $|||Q||| = 1$;
- every block matrix $A \oplus B \in \mathbb{C}^{N\times N}$ satisfies

$$|||A \oplus B||| = \max(|||A \oplus 0|||, |||0 \oplus B|||).$$

For example, if $|\cdot|_p$ is the usual $p$-norm on $\mathbb{C}^N$, given by $|x|_p := (\sum_{j=1}^N |x_j|^p)^{1/p}$, then the associated operator norm on $\mathbb{C}^{N\times N}$ is admissible.

The following result is a generalization of Theorem 1.1 in this context.

THEOREM 1.6. [illegible] $n \geq 2$ [illegible] $\alpha_2, \ldots, \alpha_n$ [illegible] $\beta_2, \ldots, \beta_n$ [illegible] $N \geq 1$ [illegible] $A, B \in \mathbb{C}^{N\times N}$ [illegible] $|||\cdot|||$ [illegible] $\mathbb{C}^{N\times N}$

$$|||(zI - A)^{-1}||| = |||(zI - B)^{-1}||| \qquad (z \in \mathbb{C}) \tag{1.12}$$

[illegible]

$$|||A^k||| = \alpha_k \quad [illegible] \quad |||B^k||| = \beta_k \qquad (k = 2, \ldots, n). \tag{1.13}$$

[illegible] $N = 2n + 3$

We conclude by remarking that there is at least one well-known norm on $\mathbb{C}^{N\times N}$ for which matrices $A, B$ with identical pseudospectra [illegible] have identical power growth. This is the Hilbert–Schmidt (or Frobenius) norm, as was shown by Greenbaum and Trefethen in [3]. We shall need their result in section 4, where more details will be given. Of course, the Hilbert–Schmidt norm is not an admissible norm in our sense; in fact it fails all three parts of the definition.

The rest of the paper is devoted to the proofs of the six theorems above.

**2. Proof of Theorem 1.1.** The proof of Theorem 1.1 is based on a construction using weighted shifts, which will also serve as a model in several other proofs to follow. It is therefore written in such a way as to be easy to adapt to other situations.

Given $\omega_1, \ldots, \omega_n > 0$, we write

$$S(\omega_1, \ldots, \omega_n) := \begin{pmatrix} 0 & \omega_1 & 0 & \ldots & 0 \\ 0 & 0 & \omega_2/\omega_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & \omega_n/\omega_{n-1} \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix}. \tag{2.1}$$

Lemma 2.1. [illegible] $\omega_1, \dots, \omega_n$ [illegible] $S = S(\omega_1, \dots, \omega_n)$ [illegible]

$$\|S^k\| = \omega_k \qquad (k = 1, \dots, n) \tag{2.2}$$

[illegible]

$$1 + \max_{1 \le k \le n} \frac{\omega_k |z|^k}{2} \le \|(I - zS)^{-1}\| \le 1 + \sum_{k=1}^{n} \omega_k |z|^k \qquad (z \in \mathbb{C}). \tag{2.3}$$

[illegible] Let $k \in \{1, \dots, n\}$. Taking $Q$ to be an appropriate cyclic permutation matrix, we have $S^k Q = \operatorname{diag}(\omega_k, \omega_{k+1}/\omega_1, \dots, \omega_n/\omega_{n-k}, 0, \dots, 0)$. Using the submultiplicativity of the sequence $\omega_1, \dots, \omega_n$, we obtain

$$\|S^k Q\| = \max(\omega_k, \omega_{k+1}/\omega_1, \dots, \omega_n/\omega_{n-k}, 0, \dots, 0) = \omega_k.$$

Since $\|Q\| = 1 = \|Q^{-1}\|$, and $\|\cdot\|$ is an algebra norm, it follows that $\|S^k\| = \omega_k$. This proves (2.2).

For the upper bound in (2.3), note that $(I - zS)^{-1} = \sum_{k=0}^{n} z^k S^k$, whence

$$\|(1 - zS)^{-1}\| = \left\| \sum_{k=0}^{n} z^k S^k \right\| \le \sum_{k=0}^{n} |z|^k \|S^k\| = 1 + \sum_{k=1}^{n} |z|^k \omega_k \qquad (z \in \mathbb{C}).$$

For the lower bound, first fix $k \in \{1, \dots, n\}$ and $z \in \mathbb{C}$. Let $Q$ be the permutation matrix that exchanges rows 2 and $k+1$, and let $P := \operatorname{diag}(1, e^{i\theta}, 0, \dots, 0)$, where $\theta = -\arg(z^k)$. Then

$$\overline{P} Q (I - zS)^{-1} Q P = \begin{pmatrix} 1 & |z|^k \omega_k \\ 0 & 1 \end{pmatrix} \oplus 0,$$

and hence

$$\|(I - zS)^{-1}\| \ge \left\| \begin{pmatrix} 1 & |z|^k \omega_k \\ 0 & 1 \end{pmatrix} \oplus 0 \right\|.$$

Conjugating by the permutation matrix that swaps the first two rows, we have

$$\left\| \begin{pmatrix} 1 & |z|^k \omega_k \\ 0 & 1 \end{pmatrix} \oplus 0 \right\| = \left\| \begin{pmatrix} 1 & 0 \\ |z|^k \omega_k & 1 \end{pmatrix} \oplus 0 \right\|.$$

Taking averages, it follows that

$$\|(I - zS)^{-1}\| \ge \left\| \begin{pmatrix} 1 & |z|^k \omega_k / 2 \\ |z|^k \omega_k / 2 & 1 \end{pmatrix} \oplus 0 \right\|.$$

Since $\|\cdot\|$ is always at least as large as the spectral radius, we deduce that

$$\|(I - zS)^{-1}\| \ge \rho \left( \begin{pmatrix} 1 & |z|^k \omega_k / 2 \\ |z|^k \omega_k / 2 & 1 \end{pmatrix} \oplus 0 \right) = 1 + \frac{|z|^k \omega_k}{2}.$$

As this holds for each $k \in \{1, \dots, n\}$ and each $z \in \mathbb{C}$, we obtain the lower bound in (2.3). $\square$

[illegible] 1.1 First, choose $\alpha_1, \beta_1$ large enough so that the sequences $\alpha_1, \dots, \alpha_n$ and $\beta_1, \dots, \beta_n$ are submultiplicative, and set

$$A_0 := S(\alpha_1, \dots, \alpha_n) \quad \text{and} \quad B_0 := S(\beta_1, \dots, \beta_n).$$

Then set $A := A_0 \oplus C_0$ and $B := B_0 \oplus C_0$, where $C_0$ is the $(n+2) \times (n+2)$ matrix defined by

$$C_0 := S(\Gamma, \gamma^2, \gamma^3, \dots, \gamma^{n+1}).$$

Here $\Gamma, \gamma$ are positive numbers to be chosen later. Note that the sequence of numbers $\Gamma, \gamma^2, \dots, \gamma^{n+1}$ is submultiplicative provided that $\Gamma \geq \gamma$. Defined in this way, $A, B$ are $(2n+3) \times (2n+3)$ matrices, and we shall show that they satisfy (1.3) and (1.4) if $\gamma, \Gamma$ are chosen suitably.

We first choose $\gamma > 0$ small enough so that $\gamma^k < \min(\alpha_k, \beta_k)$ $(k = 2, \dots, n)$. With this choice,

$$\|A^k\| = \max(\|A_0^k\|, \|C_0^k\|) = \max(\alpha_k, \gamma^k) = \alpha_k \qquad (k = 2, \dots, n),$$

and likewise $\|B^k\| = \beta_k$ $(k = 2, \dots, n)$.

It remains to choose $\Gamma$ to ensure that $A, B$ have identical pseudospectra. This will be the case provided that

$$\left\{ \begin{array}{l} \|(I - zC_0)^{-1}\| \geq \|(I - zA_0)^{-1}\| \\ \|(I - zC_0)^{-1}\| \geq \|(I - zB_0)^{-1}\| \end{array} \right. \qquad (z \in \mathbb{C}). \tag{2.4}$$

By Lemma 2.1, this will be true if

$$\max\left(\frac{\Gamma t}{2}, \frac{\gamma^{n+1} t^{n+1}}{2}\right) \geq \sum_{k=1}^{n} \alpha_k t^k \quad \text{and} \quad \max\left(\frac{\Gamma t}{2}, \frac{\gamma^{n+1} t^{n+1}}{2}\right) \geq \sum_{k=1}^{n} \beta_k t^k \quad (t \geq 0).$$

Now there exists $t_0$, depending on $\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n, \gamma$, but not on $\Gamma$, such that

$$\frac{\gamma^{n+1} t^{n+1}}{2} \geq \sum_{k=1}^{n} \alpha_k t^k \quad \text{and} \quad \frac{\gamma^{n+1} t^{n+1}}{2} \geq \sum_{k=1}^{n} \beta_k t^k \qquad (t \geq t_0).$$

Hence it will suffice that

$$\frac{\Gamma t}{2} \geq \sum_{k=1}^{n} \alpha_k t^k \quad \text{and} \quad \frac{\Gamma t}{2} \geq \sum_{k=1}^{n} \beta_k t^k \qquad (0 \leq t \leq t_0).$$

This will certainly be true provided we choose $\Gamma$ large enough. With this choice, the construction is complete. □

**3. Proof of Theorem 1.2.** The basic idea is to perturb the construction in the proof of Theorem 1.1. However, keeping track of the norms of the resolvents requires a certain amount of care. We shall need two lemmas.

LEMMA 3.1. [illegible] $V, W \in \mathbb{C}^{N \times N}$ [illegible] $V$ [illegible] $\|V - W\| < 1/(2\|V^{-1}\|)$ [illegible] $W$ [illegible]

$$\|V^{-1} - W^{-1}\| \leq 2\|V^{-1}\|^2 \|V - W\|.$$

We have

$$\|I - V^{-1}W\| = \|V^{-1}(V - W)\| \le \|V^{-1}\|\,\|V - W\| \le 1/2.$$

Therefore $V^{-1}W$ is invertible, and hence so is $W$.

We also have

$$\|V^{-1} - W^{-1}\| = \|V^{-1}(W - V)W^{-1}\| \le \|V^{-1}\|\,\|W - V\|\,\|W^{-1}\|. \tag{3.1}$$

Since $\|W - V\| \le 1/(2\|V^{-1}\|)$, it follows that $\|V^{-1} - W^{-1}\| \le \|W^{-1}\|/2$, whence $\|W^{-1}\| \le 2\|V^{-1}\|$. Substituting this back into (3.1) gives the result. □

In the next lemma, adj denotes adjugate and $\rho$ denotes spectral radius.

LEMMA 3.2. $V \in \mathbb{C}^{N\times N}$

$$\|\mathrm{adj}(V) - (-V)^{N-1}\| \le 2^N \rho(V)\|V\|^{N-2}.$$

It suffices to prove this when $V$ is invertible, since invertible matrices are dense in $\mathbb{C}^{N\times N}$.

Let $p(z)$ be the characteristic polynomial of $V$. Since $p(z) = \prod_{j=1}^{N}(z - \lambda_j)$, where $\lambda_1, \dots, \lambda_N$ are the eigenvalues of $V$, we have $p(z) = \sum_{j=0}^{N} a_j z^j$, where

$$a_N = 1, \quad a_0 = (-1)^N \det(V) \quad \text{and} \quad |a_j| \le \binom{N}{j}\rho(V)^{N-j} \quad (j = 0, \dots, N). \tag{3.2}$$

Now by the Cayley–Hamilton theorem, $p(V) = 0$. Multiplying by $V^{-1}$ and rearranging gives

$$a_0 V^{-1} + a_N V^{N-1} = -\sum_{j=1}^{N-1} a_j V^{j-1}.$$

Using (3.2), it follows that

$$\|(-1)^N \det(V)V^{-1} + V^{N-1}\| \le \sum_{j=1}^{N-1}\binom{N}{j}\rho(V)^{N-j}\|V\|^{j-1}.$$

Since $\det(V)V^{-1} = \mathrm{adj}(V)$ and $\rho(V) \le \|V\|$, we deduce that

$$\|(-1)^N \mathrm{adj}(V) + V^{N-1}\| \le \sum_{j=1}^{N-1}\binom{N}{j}\rho(V)\|V\|^{N-j-1}\|V\|^{j-1} \le 2^N\rho(V)\|V\|^{N-2},$$

whence we have the result. □

1.2 Define $A_0, B_0, C_0$ as in the proof of Theorem 1.1. The choice of $\gamma$ is the same as before, so that $\|(A_0 \oplus C_0)^k\| = \alpha_k$ and $\|(B_0 \oplus C_0)^k\| = \beta_k$ for $k = 2, \dots, n$. This time, however, we choose $\Gamma$ a little differently, stipulating that $\Gamma \ge \gamma$ and

$$\frac{\Gamma t}{2} \ge \sum_{k=1}^{n}\alpha_k t^k + 2t \qquad \text{and} \qquad \frac{\Gamma t}{2} \ge \sum_{k=1}^{n}\beta_k t^k + 2t \qquad (0 \le t \le 1/\gamma). \tag{3.3}$$

The next step is to perturb $A_0, B_0, C_0$ so as to obtain diagonalizable matrices. To this end, we fix distinct complex numbers $\zeta_1, \dots, \zeta_{n+1}$ of modulus $1/2$, and set

$$D := \mathrm{diag}(\zeta_1, \dots, \zeta_{n+1}) \qquad \text{and} \qquad D' := \mathrm{diag}(\zeta_1, \dots, \zeta_{n+1}, 0).$$

Then, for each $\delta > 0$, we define

$$A_\delta := A_0 + \delta D, \qquad B_\delta := B_0 + \delta D, \qquad \text{and} \qquad C_\delta := C_0 + \delta D'.$$

Each of $A_\delta, B_\delta, C_\delta$ has distinct eigenvalues, so $A_\delta \oplus C_\delta$ and $B_\delta \oplus C_\delta$ are diagonalizable.

By continuity, if $\delta > 0$ is chosen small enough, then

$$\Big| \|(A_\delta \oplus C_\delta)^k\| - \alpha_k \Big| < \epsilon \quad \text{and} \quad \Big| \|(B_\delta \oplus C_\delta)^k\| - \beta_k \Big| < \epsilon \qquad (k = 2, \dots, n).$$

We next show that, reducing $\delta$ if necessary, we have

$$\left\{ \begin{array}{l} \|(C_\delta - zI)^{-1}\| \geq \|(A_\delta - zI)^{-1}\| \\ \|(C_\delta - zI)^{-1}\| \geq \|(B_\delta - zI)^{-1}\| \end{array} \right. \qquad (|z| \geq \gamma). \tag{3.4}$$

By Lemma 2.1, we have

$$\|(I - wC_0)^{-1}\| - \|(I - wA_0)^{-1}\| \geq (1 + \Gamma|w|/2) - \left(1 + \sum_{k=1}^{n} |\alpha_k||w|^k\right) \qquad (w \in \mathbb{C}).$$

From our choice of $\Gamma$ in (3.3), it follows that

$$\|(I - wC_0)^{-1}\| - \|(I - wA_0)^{-1}\| \geq 2|w| \qquad (|w| \leq 1/\gamma).$$

We now apply Lemma 3.1 with $V = I - wA_0$ and $W = I - wA_\delta$. We find that, if $\delta|w|\|D\| \leq 1/(2\|(I - wA_0)^{-1}\|)$, then

$$\|(I - wA_\delta)^{-1} - (I - wA_0)^{-1}\| \leq 2\|(I - wA_0)^{-1}\|^2 \delta|w|\|D\|.$$

It follows that, if $\delta$ is chosen small enough, then

$$\|(I - wA_\delta)^{-1} - (I - wA_0)^{-1}\| \leq |w| \qquad (|w| \leq 1/\gamma).$$

Likewise, if $\delta$ is small enough, then

$$\|(I - wC_\delta)^{-1} - (I - wC_0)^{-1}\| \leq |w| \qquad (|w| \leq 1/\gamma).$$

Putting all of this together, we find that, if $\delta$ is sufficiently small, then

$$\|(I - wC_\delta)^{-1}\| \geq \|(I - wA_\delta)^{-1}\| \qquad (|w| \leq 1/\gamma),$$

from which (3.4) follows for $A$. Evidently, a similar argument applies to $B$.

The next step is to show that, by reducing $\delta$ yet further, we may ensure that

$$\left\{ \begin{array}{l} \|(C_\delta - zI)^{-1}\| \geq \|(A_\delta - zI)^{-1}\| \\ \|(C_\delta - zI)^{-1}\| \geq \|(B_\delta - zI)^{-1}\| \end{array} \right. \qquad (|z| \leq \delta). \tag{3.5}$$

For this we use Lemma 3.2. Applying this lemma with $V = A_\delta - zI$, and recalling that this is an $(n+1) \times (n+1)$ matrix, we obtain

$$\|\operatorname{adj}(A_\delta - zI) - (zI - A_\delta)^n\| \leq 2^{n+1} \rho(A_\delta - zI) \|A_\delta - zI\|^{n-1}.$$

Now $\sigma(A_\delta - zI) = \sigma(\delta D - zI)$, so

$$\rho(A_\delta - zI) = \rho(\delta D - zI) \leq \|\delta D - zI\| \leq \delta + |z|.$$

It follows that

$$\|\operatorname{adj}(A_\delta - zI) - (zI - A_\delta)^n\| \le 2^{n+1}(\delta + |z|)\|A_\delta - zI\|^{n-1}.$$

Note also that $\sup_{|z|\le\delta} \|(zI - A_\delta)^n - (-A_0)^n\| = O(\delta)$ as $\delta \to 0$. Hence there exists a constant $K$, independent of $z, \delta$, such that

$$\|\operatorname{adj}(A_\delta - zI) - (-A_0)^n\| \le K\delta \qquad (|z| \le \delta).$$

Similarly, as $C_\delta$ is an $(n+2)\times(n+2)$ matrix, there exists a constant $K'$ such that

$$\|\operatorname{adj}(C_\delta - zI) - (-C_0)^{n+1}\| \le K'\delta \qquad (|z| \le \delta).$$

Since $A_0^n \ne 0$ and $C_0^{n+1} \ne 0$, it follows that if $\delta$ is small enough, then

$$\left\{\begin{array}{l} \|\operatorname{adj}(C_\delta - zI)\| \ge \|C_0^{n+1}\|/2 \\ \|\operatorname{adj}(A_\delta - zI)\| \le 2\|A_0^n\| \end{array}\right. \qquad (|z| \le \delta).$$

Now

$$\begin{aligned} \operatorname{adj}(C_\delta - zI) &= \det(C_\delta - zI)(C_\delta - zI)^{-1}, \\ \operatorname{adj}(A_\delta - zI) &= \det(A_\delta - zI)(A_\delta - zI)^{-1}, \end{aligned}$$

and

$$\frac{\det(C_\delta - zI)}{\det(A_\delta - zI)} = \frac{\det(\delta D' - zI)}{\det(\delta D - zI)} = -z.$$

Combining these facts, we obtain that, for sufficiently small $\delta > 0$,

$$\frac{\|(C_\delta - zI)^{-1}\|}{\|(A_\delta - zI)^{-1}\|} \ge \frac{1}{|z|}\frac{\|C_0^{n+1}\|}{4\|A_0^n\|} \ge \frac{1}{\delta}\frac{\|C_0^{n+1}\|}{4\|A_0^n\|} \qquad (|z| \le \delta).$$

Thus, if $\delta$ is chosen small enough, then (3.5) holds for $A$. The argument for $B$ is similar.

Fix $\delta > 0$ so that (3.4) and (3.5) hold. Summarizing what we have achieved so far, if we define $A := A_\delta \oplus C_\delta$ and $B := B_\delta \oplus C_\delta$, then $A, B$ are diagonalizable matrices satisfying (1.6) and

$$\|(A - zI)^{-1}\| = \|(B - zI)^{-1}\| \qquad (z \in \mathbb{C} \setminus Q),$$

where $Q$ is the annulus $\{z \in \mathbb{C} : \delta < |z| < \gamma\}$. Our remaining task is to deal with the case $z \in Q$, which we do as follows. Let $L$ be the maximum of $\sup_{z\in Q} \|(A - zI)^{-1}\|$ and $\sup_{z\in Q} \|(B - zI)^{-1}\|$. Cover $Q$ by a finite number of disks of radius $1/L$, with centers $\mu_1, \dots, \mu_m \in Q$, say. Define $E := \operatorname{diag}(\mu_1, \dots, \mu_m)$. Then we have

$$\left\{\begin{array}{l} \|(E - zI)^{-1}\| \ge \|(A - zI)^{-1}\| \\ \|(E - zI)^{-1}\| \ge \|(B - zI)^{-1}\| \end{array}\right. \qquad (z \in Q).$$

Thus, if we replace $A, B$ with $A \oplus E, B \oplus E$, respectively, then they have identical pseudospectra. Evidently the new $A, B$ are still diagonalizable. Finally, as $\mu_1, \dots, \mu_m \in Q$, we have $\|E^k\| \le \gamma^k \le \min(\alpha_k, \beta_k)$ for $k = 2, \dots, n$, and so (1.6) still holds. The construction is complete. $\square$

[illegible] The construction yields matrices $A, B$ having eigenvalues of multiplicity at most two. It would be interesting to obtain an example where the eigenvalues were all of multiplicity one.

**4. Proofs of Theorems 1.3 and 1.4.** Theorem 1.3 is an easy consequence of the following result of Greenbaum and Trefethen. Recall that the Hilbert–Schmidt norm of a square matrix $A$ is defined by

$$\|A\|_{HS} := \sqrt{\operatorname{trace}(A^*A)}.$$

THEOREM 4.1 (see [3, Theorem 3]). [illegible] $A, B \in \mathbb{C}^{N\times N}$ [illegible]

$$\|(zI - A)^{-1}\|_{HS} = \|(zI - B)^{-1}\|_{HS} \qquad (z \in \mathbb{C}). \tag{4.1}$$

[illegible] $p$

$$\|p(A)\|_{HS} = \|p(B)\|_{HS}. \tag{4.2}$$

Since [3] was never published, we also include a brief proof for the reader's convenience.

[illegible] Setting $\zeta = 1/z$, we see that (4.1) is equivalent to

$$\operatorname{trace}[(I - \overline{\zeta}A^*)^{-1}(I - \zeta A)^{-1}] = \operatorname{trace}[(I - \overline{\zeta}B^*)^{-1}(I - \zeta B)^{-1}] \qquad (\zeta \in \mathbb{C}). \tag{4.3}$$

Expanding, we deduce that, for some $r > 0$,

$$\sum_{k,l\geq 0} \operatorname{trace}(A^{*k}A^l)\overline{\zeta}^k\zeta^l = \sum_{k,l\geq 0} \operatorname{trace}(B^{*k}B^l)\overline{\zeta}^k\zeta^l \qquad (|\zeta| < r).$$

Taking $\left(\frac{\partial}{\partial\overline{\zeta}}\right)^k\left(\frac{\partial}{\partial\zeta}\right)^l$ of both sides and then setting $\zeta = 0$, we obtain

$$\operatorname{trace}(A^{*k}A^l) = \operatorname{trace}(B^{*k}B^l) \qquad (k, l \geq 0).$$

Now, let $p$ be a polynomial, say $p(z) = \sum_{j=0}^n a_j z^j$. Then

$$\begin{aligned}\operatorname{trace}(p(A)^*p(A)) &= \sum_{k,l=0}^n \overline{a}_k a_l \operatorname{trace}(A^{*k}A^l)\\ &= \sum_{k,l=0}^n \overline{a}_k a_l \operatorname{trace}(B^{*k}B^l)\\ &= \operatorname{trace}(p(B)^*p(B)),\end{aligned}$$

whence we have $\|p(A)\|_{HS} = \|p(B)\|_{HS}$. This completes the proof. □

[illegible] 1.3 Observe that $\|A\|_{HS}^2 = \sum_{j=1}^N s_j(A)^2$. Thus, hypothesis (1.7) implies that (4.1), and consequently also (4.2), holds. For each polynomial $p$, we therefore have

$$\sum_{j=1}^N s_j(p(A))^2 = \sum_{j=1}^N s_j(p(B))^2.$$

Recalling that the usual operator norm $\|\cdot\|$ is just the first singular value $s_1$, we thus obtain

$$\|p(A)\|^2 = s_1(p(A))^2 \leq \sum_{j=1}^N s_j(p(A))^2 = \sum_{j=1}^N s_j(p(B))^2 \leq Ns_1(p(B))^2 = N\|p(B)\|^2.$$

This gives the right-hand side of (1.8), and the left-hand side is proved similarly. □

[illegible] In going from (1.7) to (4.1), we are losing some information. In fact (1.7) is equivalent to the following, more complicated version of (4.3):

$$\operatorname{trace}\Big([(I-\overline{\zeta}A^*)^{-1}(I-\zeta A)^{-1}]^n\Big) = \operatorname{trace}\Big([(I-\overline{\zeta}B^*)^{-1}(I-\zeta B)^{-1}]^n\Big) \quad (\zeta \in \mathbb{C},\ n \geq 1).$$

Until now, we have not seen how to exploit this.

[illegible] 1.4 We repeat the construction in the proof of Theorem 1.1, defining $A_0, B_0, C_0$ exactly as in that proof. This time, however, we define

$$A := A_0 \oplus \overbrace{C_0 \oplus \cdots \oplus C_0}^{m} \qquad \text{and} \qquad B := B_0 \oplus \overbrace{C_0 \oplus \cdots \oplus C_0}^{m}.$$

Then $A, B \in \mathbb{C}^{N \times N}$, where $N = (n+1) + m(n+2) = (m+1)(n+2) - 1$. Just as before, we have

$$\|A^k\| = \max(\|A_0^k\|, \|C_0^k\|, \ldots, \|C_0^k\|) = \alpha_k \qquad (k = 2, \ldots, n),$$

which holds similarly for $B$, so (1.10) holds. Also, since

$$(zI - A)^{-1} = (zI - A_0)^{-1} \oplus \overbrace{(zI - C_0)^{-1} \oplus \cdots \oplus (zI - C_0)^{-1}}^{m},$$

and $\|(zI - C_0)^{-1}\| \geq \|(zI - A_0)^{-1}\|$ for all $z \in \mathbb{C}$ (see (2.4)), it follows that

$$s_j\Big((zI - A)^{-1}\Big) = \|(zI - C_0)^{-1}\| \qquad (z \in \mathbb{C},\ j = 1, \ldots, m).$$

Likewise, the same is true when $A$ is replaced with $B$. Thus (1.9) holds, and the proof is complete. □

## 5. Proofs of Theorems 1.5 and 1.6.

We shall in fact prove the following slight generalization of Theorem 1.5.

Theorem 5.1. [illegible] $A, B \in \mathbb{C}^{N \times N}$ [illegible]

$$\|(I - \zeta A)^{-1}\|_1 = \|(I - \zeta B)^{-1}\|_1 + o(\zeta) \qquad \text{[illegible]} \quad \zeta \to 0,\ \zeta \in \mathbb{C}.$$

[illegible] $\|A\|_1 = \|B\|_1$

[illegible] The norm $\|\cdot\|_1$ has the particularity that $\|A\|_1 = \max(|Ae_1|_1, \ldots, |Ae_N|_1)$, where $e_1, \ldots, e_N$ is the standard unit vector basis of $\mathbb{C}^N$. Fix a $j$ so that $\|A\|_1 = |Ae_j|_1$. Multiplying $A$ and $B$ by the same unimodular constant, we may suppose that $a_{jj} \geq 0$; in other words, the $j$th entry in $Ae_j$ is nonnegative. It then follows that, for all $t \geq 0$,

$$|(I + tA)e_j|_1 = 1 + t|Ae_j|_1 = 1 + t\|A\|_1.$$

On the other hand, as $t \to 0^+$, we have

$$\begin{aligned} |(I + tA)e_j|_1 \leq \|I + tA\|_1 &= \|(I - tA)^{-1}\|_1 + o(t) \\ &= \|(I - tB)^{-1}\|_1 + o(t) \leq 1 + t\|B\|_1 + o(t). \end{aligned}$$

Combining these facts, we deduce that $\|A\|_1 \leq \|B\|_1$. By symmetry, $\|B\|_1 \leq \|A\|_1$ as well. □

[illegible] This theorem may be viewed as a result about numerical ranges in Banach algebras. For background on numerical ranges, we refer to [1, 2]. Let $(\mathcal{A}, \|\cdot\|_{\mathcal{A}})$ be a Banach algebra with identity 1, and given $a \in \mathcal{A}$, let $\nu_{\mathcal{A}}(a)$ denote the numerical radius of $a$. It is well known that

$$\nu_{\mathcal{A}}(a) = \limsup_{\zeta \to 0} \frac{\|1 + \zeta a\|_{\mathcal{A}} - 1}{|\zeta|} \qquad (a \in \mathcal{A}),$$

and also that there exists a constant $n(\mathcal{A}) \in [e^{-1}, 1]$, called the [illegible] of $\mathcal{A}$, such that

$$n(\mathcal{A})\|a\|_{\mathcal{A}} \le \nu_{\mathcal{A}}(a) \le \|a\|_{\mathcal{A}} \qquad (a \in \mathcal{A}).$$

From these facts it follows easily that, if $a, b \in \mathcal{A}$ satisfy

$$\|(1 - \zeta a)^{-1}\| = \|(1 - \zeta b)^{-1}\| + o(\zeta) \qquad \text{as } \zeta \to 0,\ \zeta \in \mathbb{C},$$

then $\nu_{\mathcal{A}}(a) = \nu_{\mathcal{A}}(b)$, and hence

$$n(\mathcal{A}) \le \|a\|_{\mathcal{A}}/\|b\|_{\mathcal{A}} \le n(\mathcal{A})^{-1}.$$

Moreover, it is known that the numerical indices of $(\mathbb{C}^{N\times N}, \|\cdot\|)$ and $(\mathbb{C}^{N\times N}, \|\cdot\|_1)$ are equal to 1/2 and 1, respectively. We thus recover as special cases both the result (1.2) mentioned earlier and Theorem 5.1 above.

We now turn to the proof of Theorem 1.6. Recall that the notion of admissible norm on $\mathbb{C}^{N\times N}$ was defined in the introduction, and that the weighted shift $S(\omega_1, \ldots, \omega_n)$ was defined in (2.1).

LEMMA 5.2. [illegible] $\omega_1, \ldots, \omega_n$ [illegible] $S = S(\omega_1, \ldots, \omega_n)$ [illegible] $|||\cdot|||$ [illegible] $\mathbb{C}^{(n+1)\times(n+1)}$ [illegible]

$$|||S^k||| = \omega_k \qquad (k = 1, \ldots, n)$$

[illegible]

$$1 + \max_{1\le k\le n} \frac{\omega_k |z|^k}{2} \le |||(I - zS)^{-1}||| \le 1 + \sum_{k=1}^{n} \omega_k |z|^k \qquad (z \in \mathbb{C}).$$

[illegible] Repeat the proof of Lemma 2.1, observing that it is valid for every admissible norm. □

[illegible] 1.6 Repeat the proof of Theorem 1.1, using Lemma 5.2 in place of Lemma 2.1. Note that the choices of $\gamma$ and $\Gamma$ depend only on the $\alpha_j$ and $\beta_j$, and not on the particular norm. Thus, the same pair of matrices $A, B$ works simultaneously for all admissible norms $|||\cdot|||$. □

**Acknowledgments.** I am greatly indebted to Nick Trefethen for introducing me to this topic, for making available the article [3], and for numerous invaluable discussions. This work was carried out while I was visiting the Mathematical Institute of the University of Oxford and the Oxford University Computing Laboratory, and I am grateful to both institutions for their hospitality.

## REFERENCES

[1] F. F. Bonsall and J. Duncan, *Numerical Ranges of Operators on Normed Spaces and of Elements of Normed Algebras*, Cambridge University Press, Cambridge, UK, 1971.

[2] F. F. Bonsall and J. Duncan, *Numerical Ranges* II, Cambridge University Press, Cambridge, UK, 1973.

[3] A. Greenbaum and L. N. Trefethen, *Do the Pseudospectra of a Matrix Determine Its Behavior?*, Technical Report TR 93-1371, Computer Science Department, Cornell University, Ithaca, NY, 1993.

[4] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra*, Princeton University Press, Princeton, NJ, 2005.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 712–730

# MEAN-SQUARED ERROR ESTIMATION FOR LINEAR SYSTEMS WITH BLOCK CIRCULANT UNCERTAINTY*

AMIR BECK†, YONINA C. ELDAR‡, AND AHARON BEN-TAL§

**Abstract.** We consider the problem of estimating a vector $\mathbf{x}$ in the linear model $\mathbf{Ax} \approx \mathbf{y}$, where $\mathbf{A}$ is a block circulant (BC) matrix with $N$ blocks and $\mathbf{x}$ is assumed to have a weighted norm bound. In the case where both $\mathbf{A}$ and $\mathbf{y}$ are subjected to noise, we propose a minimax mean-squared error (MSE) approach in which we seek the linear estimator that minimizes the worst-case MSE over a BC structured uncertainty region. For an arbitrary choice of weighting, we show that the minimax MSE estimator can be formulated as a solution to a semidefinite programming problem (SDP), which can be solved efficiently. For a Euclidean norm bound on $\mathbf{x}$, the SDP is reduced to a simple convex program with $N+1$ unknowns. Finally, we demonstrate through an image deblurring example the potential of the minimax MSE approach in comparison with other conventional methods.

**Key words.** minimax estimation, block circulant structure, semidefinite programming, robust optimization

**AMS subject classifications.** 90C22, 65F30, 90C90

**DOI.** 10.1137/050643696

## 1. Introduction.

Many problems in data fitting and estimation give rise to a system of linear equations $\mathbf{Ax} \approx \mathbf{y}$, where both the matrix $\mathbf{A}$ and the right-hand side $\mathbf{y}$ are contaminated by noise. Given the observation $\mathbf{y}$, we seek an estimator $\hat{\mathbf{x}}$ of $\mathbf{x}$ that is close in some sense to $\mathbf{x}$. This estimation problem arises in a large variety of areas in science and engineering, e.g., communication, economics, signal processing, seismology, and control.

Several approaches for dealing with uncertainties in the model matrix $\mathbf{A}$ and right-hand side vector $\mathbf{y}$ are known in the literature. In the *total least-squares* (TLS) strategy [11, 15], one seeks the minimal norm perturbations $\mathbf{\Delta A}, \mathbf{\Delta y}$ of the nominal model matrix $\mathbf{A}$ and observation vector $\mathbf{y}$ such that the linear system $(\mathbf{A}+\mathbf{\Delta A})\mathbf{x} = \mathbf{y}+\mathbf{\Delta y}$ is consistent. An alternative strategy is the *robust least-squares* (RLS) method [10, 22, 6]. Here the underlying assumption is that the perturbation matrix $\mathbf{\Delta A}$ and the perturbation vector $\mathbf{\Delta y}$ belong to some bounded uncertainty set $\mathcal{U}$. The solution (or estimator) is chosen to minimize the worst-case data error (or "residual") over the uncertainty region:

$$\hat{\mathbf{x}}_{\mathrm{RLS}} \in \operatorname*{argmin}_{\mathbf{x}} \max_{(\mathbf{\Delta A},\mathbf{\Delta y})\in\mathcal{U}} \|(\mathbf{A}+\mathbf{\Delta A})\mathbf{x} - \mathbf{y} - \mathbf{\Delta y}\|^2. \tag{1.1}$$

Both the RLS and TLS solutions optimize a criterion that is based on the *data error*

---

[*] Received by the editors October 28, 2005; accepted for publication (in revised form) by L. Vandenberghe January 2, 2007; published electronically June 5, 2007.
http://www.siam.org/journals/simax/29-3/64369.html

[†] MINERVA Optimization Center, Department of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel (becka@ie.technion.ac.il). The research of this author was partially supported by the Israel Science Foundation under grant 729/04.

[‡] Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (yonina@ee.technion.ac.il). The research of this author was partially supported by the Israel Science Foundation under grant 536/04 and by the EU 6th framework programme via the NEWCOM network of excellence.

[§] MINERVA Optimization Center, Department of Industrial Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (abental@ie.technion.ac.il). The research of this author was partially supported by the Israel Science Foundation under grant 729/04.

*residual* ($\|\mathbf{A}\mathbf{x}-\mathbf{y}\|$ or $\|(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A})\mathbf{x}-\mathbf{y}-\boldsymbol{\Delta}\mathbf{y}\|$) and therefore might provide poor solutions in terms of the *estimation error* $\|\mathbf{x}-\hat{\mathbf{x}}\|$. In view of this, the work [8] suggests seeking an estimator $\hat{\mathbf{x}}$ that minimizes the *mean-squared error* (MSE):

$$\text{MSE} = E(\|\mathbf{x}-\hat{\mathbf{x}}\|^2),$$

and restricting attention to *linear estimators* of the form $\hat{\mathbf{x}} = \mathbf{G}\mathbf{y}$. The expectation is with respect to the noise vector $\boldsymbol{\Delta}\mathbf{y}$, which is assumed to have a zero mean and a positive definite covariance matrix $\mathbf{C}$. For a linear estimator, the MSE is equal to the sum of the variance $V(\hat{\mathbf{x}})$ and the squared norm of the bias $B(\hat{\mathbf{x}})$:

$$E(\|\mathbf{x}-\hat{\mathbf{x}}\|^2) = \underbrace{\operatorname{Tr}(\mathbf{G}\mathbf{C}\mathbf{G}^*)}_{V(\hat{\mathbf{x}})} + \underbrace{\mathbf{x}^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}}_{\|B(\hat{\mathbf{x}})\|^2}.$$

Since the bias depends on the unknown vector $\mathbf{x}$ and the unknown perturbation matrix $\boldsymbol{\Delta}\mathbf{A}$, we cannot choose an estimator to directly minimize the MSE. The approach advocated in [8, 7], in order to minimize the MSE, is to use additional a priori information on the vector $\mathbf{x}$, such as an upper bound on its weighted norm, $\mathbf{x}^*\mathbf{T}\mathbf{x} \le L^2$, where $\mathbf{T}$ is a positive definite matrix, and minimize the worst-case MSE. This leads to the following optimization problem:

$$\min_{\mathbf{G}} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}} E(\|\mathbf{x}-\hat{\mathbf{x}}\|^2), \tag{1.2}$$

where $\mathcal{U}$ is an uncertainty set associated with the matrix $\mathbf{A}$. The optimal solution $\mathbf{G}$ of the latter problem is called the *minimax MSE matrix*, and the associated linear estimator $\hat{\mathbf{x}} = \mathbf{G}\mathbf{y}$ is termed the *minimax MSE estimator*. In the case when $\mathcal{U}$ is given by a single norm bound, it was shown in [8] that the optimal $\mathbf{G}$ can be obtained by solving a semidefinite programming (SDP) problem. In practice, if $L$ is unknown, then we can estimate it from the data, for example by using the LS estimator [3].

In this paper we study the minimax MSE estimator when the matrix $\mathbf{A}$ has a *block circulant* (BC) structure:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \mathbf{A}_{N-1} \\ \mathbf{A}_{N-1} & \mathbf{A}_0 & \cdots & \mathbf{A}_{N-2} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_0 \end{pmatrix}, \tag{1.3}$$

where $\mathbf{A}_j \in \mathbb{C}^{n\times m}$, $0 \le j \le N-1$. We use the notation $\mathbf{A} = \mathcal{C}(\mathbf{A}_0,\dots,\mathbf{A}_{N-1})$ for brevity. The BC structure of $\mathbf{A}$ imposes the same structure on the perturbation matrix, i.e., $\boldsymbol{\Delta}\mathbf{A} = \mathcal{C}(\boldsymbol{\Delta}\mathbf{A}_0,\dots,\boldsymbol{\Delta}\mathbf{A}_{N-1})$ with $\boldsymbol{\Delta}\mathbf{A}_j \in \mathbb{C}^{n\times m}$. We also assume that both the covariance matrix $\mathbf{C}$ and weighting matrix $\mathbf{T}$ are positive definite BC (which includes the case $\mathbf{C} = \sigma^2\mathbf{I}$ and $\mathbf{T} = \mathbf{I}$). Thus, the optimization problem we consider is

$$\min_{\mathbf{G}} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \left\{\operatorname{Tr}(\mathbf{G}\mathbf{C}\mathbf{G}^*) + \mathbf{x}^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}\right\}, \tag{1.4}$$

where the set $\mathcal{U}_\Delta$, which is the set of possible values of $\boldsymbol{\Delta}\mathbf{A}$, is given by

$$\mathcal{U}_\Delta \overset{\triangle}{=} \{\boldsymbol{\Delta} = \mathcal{C}(\boldsymbol{\Delta}_0,\dots,\boldsymbol{\Delta}_{N-1}) : \|\boldsymbol{\Delta}_k\| \le \rho_k,\ 0 \le k \le N-1\}. \tag{1.5}$$

Here $\|\mathbf{M}\|$ denotes the Frobenius norm of $\mathbf{M}$.

The BC model has previously been used in a variety of signal processing problems, including image restoration [17], cyclic convolution filter banks [20], texture synthesis and recognition [24], and detection techniques for CDMA systems [26]. Moreover, in many practical scenarios $\mathbf{A}$ is a block Toeplitz matrix which can be approximated by a BC matrix [12, 9]. We refer the reader to the example in section 5 that describes a usage of this Toeplitz/circulant approximation in an image deblurring context. The BC structure also includes the multiple observation model in which the matrix $\mathbf{A}$ is a block diagonal matrix with the same diagonal matrix (corresponding to $\mathbf{A}_1 = \mathbf{A}_2 = \cdots = \mathbf{A}_{N-1} = \mathbf{0}$ in (1.3)). The minimax MSE estimator for the multiple observation model was studied in [2].

Besides including several cases of practical interest, one of the attributes of the BC structure is its analytical tractability. In fact the minimax MSE problem (1.4) is intractable for most choices of uncertainty sets $\mathcal{U}_\Delta$. However, in the BC model we are able to exploit properties of BC matrices (in particular, the matrix discrete Fourier transform (DFT)) that will enable us to develop a computationally tractable scheme for computing the minimax MSE estimator.

The BC model has been investigated in the context of structured TLS problems in [1], where it was shown that by using the matrix DFT, the problem can be decomposed into several unstructured TLS problems.

The paper is organized as follows. We begin by reviewing in section 2 some properties of BC matrices and the matrix DFT. In section 3 we first show that under the BC model, the optimal minimax MSE estimator $\hat{\mathbf{x}} = \mathbf{G}\mathbf{y}$ is such that $\mathbf{G}$ is a BC matrix. This allows us to formulate the minimax MSE estimator as a solution to an SDP, which is a tractable (i.e., polynomial solvable) convex optimization problem that can be solved, e.g., using interior point methods [21, 25, 4]. In section 4 we treat the case where the weighting matrix $\mathbf{T}$ is the identity matrix $\mathbf{I}$. When the matrix $\mathbf{A}$ is [illegible], we derive an explicit formula for the minimax MSE estimator. When $\mathbf{A}$ is [illegible] but the noise vector consists of independent and identically distributed random variables ($\mathbf{C} = \sigma^2\mathbf{I}$), we show that the task of computing the minimax MSE estimator reduces to solving a simple convex program in $N+1$ variables. Finally, we demonstrate through an image deblurring example, in section 5, the potential of the minimax MSE approach in comparison with other conventional strategies.

[illegible]. We denote vectors by boldface lowercase letters and matrices by boldface uppercase letters. The identity matrix of appropriate dimension is denoted by $\mathbf{I}$, $(\cdot)^*$ and $(\cdot)^T$ denote the Hermitian conjugate and the transpose of the corresponding matrices, respectively, and $\hat{(\cdot)}$ denotes an estimated vector. For two Hermitian matrices $\mathbf{A}, \mathbf{B}$, the notation $\mathbf{A} \succeq \mathbf{B}$ means that $\mathbf{A} - \mathbf{B}$ is a positive semidefinite matrix. For a Hermitian matrix $\mathbf{A}$, $\lambda_{\max}(\mathbf{A})$ denotes the largest eigenvalue of $\mathbf{A}$. We denote by $\|\mathbf{v}\|$ the Euclidean norm of the vector $\mathbf{v}$ and by $\|\mathbf{A}\| = \sqrt{\mathrm{Tr}(\mathbf{A}^*\mathbf{A})}$ the Frobenius norm of the matrix $\mathbf{A}$. For a given matrix $\mathbf{M}$, $\mathbf{m} = \mathrm{vec}(\mathbf{M})$ denotes the vector obtained by stacking the columns of $\mathbf{M}$.

## 2. BC matrices and the DFT.

The aim of this short section is to give a brief review of results on BC matrices and the DFT defined on them that will be used later in the paper. These results can also be found in [1, 2], and they are presented here for completeness.

We begin by noting that the result of multiplication, addition, and conjugation of BC matrices is also a BC matrix. Let $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$; then the DFT of $\mathbf{A}$ is also a BC matrix of the same dimensions given by

$$\mathbf{F}(\mathbf{A}) = \mathcal{C}(\mathbf{F}_0(\mathbf{A}), \mathbf{F}_1(\mathbf{A}), \dots, \mathbf{F}_{N-1}(\mathbf{A})),$$

where $\mathbf{F}_j(\mathbf{A})$ are defined as

$$\mathbf{F}_j(\mathbf{A}) \triangleq \sum_{k=0}^{N-1} \omega^{kj}\mathbf{A}_k, \quad 0 \le j \le N-1,$$

with $\omega = e^{-\frac{2\pi i}{N}}$ (here $i = \sqrt{-1}$). The matrices $\mathbf{F}_j(\mathbf{A})$ are called the [illegible]. The inverse DFT, denoted by $\mathbf{F}^{-1}$, is defined by $\mathbf{F}^{-1}(\mathbf{A}) = (\mathbf{F}_0^{-1}(\mathbf{A}), \mathbf{F}_1^{-1}(\mathbf{A}), \dots, \mathbf{F}_{N-1}^{-1}(\mathbf{A}))$, where

$$\mathbf{F}_j^{-1}(\mathbf{A}) = \frac{1}{N}\sum_{k=0}^{N-1} \omega^{-kj}\mathbf{A}_k, \quad 0 \le j \le N-1.$$

Note that $\mathbf{F}^{-1}$ is indeed an inverse of $\mathbf{F}$ in the sense that for every BC matrix $\mathbf{A}$

$$\mathbf{F}^{-1}(\mathbf{F}(\mathbf{A})) = \mathbf{A}, \quad \mathbf{F}(\mathbf{F}^{-1}(\mathbf{A})) = \mathbf{A}.$$

The following properties of $\mathbf{F}_j$ are generalizations of well-known properties of the DFT.

LEMMA 2.1. [illegible] $\mathbf{A}$ $\mathbf{B}$ [illegible] $\mathbf{C}$ [illegible] $0 \le j \le N-1$ [illegible]

1. $(\mathbf{F}_j(\mathbf{A}))^* = \mathbf{F}_j(\mathbf{A}^*)$
2. $\mathbf{F}_j(\mathbf{I}_{mN}) = \mathbf{I}_m$
3. $\mathbf{F}_j(\mathbf{A}+\mathbf{C}) = \mathbf{F}_j(\mathbf{A}) + \mathbf{F}_j(\mathbf{C})$
4. $\mathbf{F}_j(\mathbf{AB}) = \mathbf{F}_j(\mathbf{A})\mathbf{F}_j(\mathbf{B})$
5. [illegible] $\mathbf{A}$ [illegible] $\mathbf{F}_j(\mathbf{A}^{-1}) = (\mathbf{F}_j(\mathbf{A}))^{-1}$

Theorem 2.1 shows that the eigenvalues of a Hermitian BC matrix are exactly the eigenvalues of its discrete Fourier components. Theorem 2.1 below is an extension of a well-known result on circulant matrices to the case of Hermitian block circulant matrices; for a proof, see, e.g., [2, Theorem A.1].

THEOREM 2.1. [illegible] $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1} \in \mathbb{C}^{k\times k}$ [illegible] $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1})$ [illegible] $0 \le j \le N-1$ [illegible] $\lambda_{j,0}, \lambda_{j,1}, \dots, \lambda_{j,k-1}$ [illegible] $\mathbf{F}_j(\mathbf{A})$ [illegible] $\mathbf{A}$ [illegible] $N \cdot k$ [illegible] $\lambda_{j,i}$ $0 \le i \le k-1$ $0 \le j \le N-1$

## 3. Minimax MSE estimator for BC systems.

We now use the properties of BC matrices and the DFT discussed in the previous section in order to find a $\mathbf{G}$ which is a solution to (1.4). Section 3.1 establishes the fact that $\mathbf{G}$ can always be chosen as a BC matrix. In section 3.2 we use this structure of $\mathbf{G}$ to find an SDP formulation of the estimation problem (1.4), where an SDP is the problem of minimizing a linear objective subject to linear matrix inequality (LMI) constraints, i.e., constraints of the form $\mathcal{B}(\mathbf{x}) \succeq 0$, where the matrix $\mathcal{B}$ depends linearly on $\mathbf{x}$. The advantage in this formulation is that it readily lends itself to efficient computational methods. Indeed, by exploiting the many well-known algorithms for solving SDPs, e.g., interior point methods [21, 25, 23], the optimal estimator can be computed efficiently in polynomial time. Furthermore, SDP-based algorithms are guaranteed to converge to the global optimum.

### 3.1. The structure of G.

Before proceeding, we introduce some notation. The set of all permutations of $\{0, 1, \dots, N-1\}$ is denoted by $S_N$. For every permutation

$\sigma \in S_N$ and a positive integer $l$, we associate an $lN \times lN$ matrix $\mathbf{P}_{\sigma,l}$ comprised of $N \times N$ blocks of size $l \times l$. The $(k, j)$ block of $\mathbf{P}_{\sigma,l}$ is defined as

$$(\mathbf{P}_{\sigma,l})_{k,j} = \delta_{j,\sigma(k)}\mathbf{I}_l,$$

where

$$\delta_{k,j} = \begin{cases} 0, & k \neq j, \\ 1, & k = j \end{cases}$$

is the Kronecker delta. For example, if $N = 3$ and $\sigma(0) = 1$, $\sigma(1) = 0$, and $\sigma(2) = 2$, then

$$\mathbf{P}_{\sigma,3} = \begin{pmatrix} \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{pmatrix},$$

where $\mathbf{I}_3$ is the identity matrix of size $3 \times 3$. We will be interested particularly in a special class of permutations,

$$\mathcal{A} = \{\sigma_0, \sigma_1, \ldots, \sigma_{N-1}\},$$

where $\sigma_k(j) = (j + k) \bmod N$. For example, if $N = 3$, then

$$\mathbf{P}_{\sigma_0,3} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \end{pmatrix}, \quad \mathbf{P}_{\sigma_2,3} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} \end{pmatrix}.$$

Permutation matrices $\mathbf{P}_{\sigma,l}$ satisfy some interesting properties that will be useful later on in the proof of Theorem 3.1.

[illegible] A. For every $\sigma \in S_N$ and positive integer $l$, $\mathbf{P}_{\sigma,l}\mathbf{P}^*_{\sigma,l} = \mathbf{P}^*_{\sigma,l}\mathbf{P}_{\sigma,l} = \mathbf{I}$.

[illegible] B. For every BC matrix $\mathbf{A} = \mathcal{C}(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_{N-1})$, where $\mathbf{A}_k \in \mathbb{C}^{m,n}$, and every permutation $\sigma$ in the class $\mathcal{A}$, we have that $\mathbf{P}_{\sigma,m}\mathbf{A}\mathbf{P}^*_{\sigma,n} = \mathbf{A}$ or, equivalently, $\mathbf{P}_{\sigma,m}\mathbf{A} = \mathbf{A}\mathbf{P}_{\sigma,n}$.

The main result of this section is presented in Theorem 3.1, where we show that the solution of (1.4) is a BC matrix, i.e., $\mathbf{G} = \mathcal{C}(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{N-1})$ for some $\mathbf{G}_0, \ldots, \mathbf{G}_{N-1} \in \mathbb{C}^{m \times n}$.

THEOREM 3.1. [illegible] $\mathbf{x}$ [illegible] $\mathbf{y} = (\mathbf{A} + \mathbf{\Delta A})\mathbf{x} + \mathbf{\Delta y}$ [illegible] $\mathbf{A}$ [illegible] $\mathbf{\Delta A}$ [illegible] $\mathbf{\Delta A} \in \mathcal{U}_\Delta$ [illegible] $\mathcal{U}_\Delta$ [illegible] (1.5) [illegible] $\mathbf{\Delta y}$ [illegible] $\mathbf{C}$ [illegible] $\mathbf{T}$ [illegible]

$$\min_{\mathbf{G}} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x} \le L^2,\, \Delta\mathbf{A} \in \mathcal{U}_\Delta} E(\|\hat{\mathbf{x}} - \mathbf{x}\|^2)$$

[illegible] $\mathbf{G}$ [illegible]

[illegible]. We first rewrite problem (1.2) as

$$\min_{\mathbf{G} \in \mathbb{C}^{m \times n}} \Gamma(\mathbf{G}), \tag{3.1}$$

where

$$\Gamma(\mathbf{G}) = \max_{\mathbf{x}^*\mathbf{T}\mathbf{x} \le L^2,\, \Delta\mathbf{A} \in \mathcal{U}_\Delta} E(\|\hat{\mathbf{x}} - \mathbf{x}\|^2).$$

The function $\Gamma$ can be decomposed as follows (see (1.4)):

$$\Gamma(\mathbf{G}) = \theta_1(\mathbf{G}) + \theta_2(\mathbf{G}),$$

where

$$\begin{aligned}
\theta_1(\mathbf{G}) &= \operatorname{Tr}(\mathbf{G}\mathbf{C}\mathbf{G}^*),\\
\theta_2(\mathbf{G}) &= \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \varphi(\mathbf{G},\mathbf{x},\boldsymbol{\Delta}\mathbf{A}),
\end{aligned}$$

and $\varphi(\mathbf{G},\mathbf{x},\boldsymbol{\Delta}\mathbf{A}) \overset{\triangle}{=} \mathbf{x}^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}$. It is easy to see that the positive definiteness of $\mathbf{C}$ implies strict convexity of $\theta_1$. Moreover, since $\varphi$ is a convex function with respect to $\mathbf{G}$, it follows that $\theta_2$, being a maximum of convex functions, is also a convex function. Thus, $\Gamma = \theta_1 + \theta_2$ is a strictly convex function, and hence it has a unique optimal solution.

Using Properties A and B, we have

$$\begin{aligned}
\theta_1(\mathbf{G}) = \operatorname{Tr}(\mathbf{G}\mathbf{C}\mathbf{G}^*) &\overset{A}{=} \operatorname{Tr}(\mathbf{P}^*_{\sigma,m}\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{C}\mathbf{G}^*)\\
&= \operatorname{Tr}(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{C}\mathbf{G}^*\mathbf{P}^*_{\sigma,m})\\
&\overset{B}{=} \operatorname{Tr}(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n}\mathbf{C}\mathbf{P}_{\sigma,n}\mathbf{G}^*\mathbf{P}^*_{\sigma,m})\\
&= \operatorname{Tr}((\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n})\mathbf{C}(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n})^*)\\
&= \theta_1(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n})
\end{aligned}$$

and

$$\begin{aligned}
\theta_2(\mathbf{G}) &= \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \{\mathbf{x}^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}\}\\
&= \max_{\mathbf{x}^*\mathbf{P}^*_{\sigma,m}\mathbf{T}\mathbf{P}_{\sigma,m}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \{\mathbf{x}^*\mathbf{P}^*_{\sigma,m}(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{P}_{\sigma,m}\mathbf{x}\}\\
&\overset{B}{=} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \{\mathbf{x}^*\mathbf{P}^*_{\sigma,m}(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{P}_{\sigma,m}\mathbf{x}\}\\
&\overset{A}{=} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \{\mathbf{x}^*\mathbf{P}^*_{\sigma,m}(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*\mathbf{P}_{\sigma,m}\mathbf{P}^*_{\sigma,m}(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{P}_{\sigma,m}\mathbf{x}\}\\
&\overset{A}{=} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \{\mathbf{x}^*(\mathbf{I}-\mathbf{P}_{\sigma,m}\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A})\mathbf{P}^*_{\sigma,m})^*(\mathbf{I}-\mathbf{P}_{\sigma,m}\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A})\mathbf{P}^*_{\sigma,m})\mathbf{x}\}\\
&\overset{B}{=} \max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\, \Delta\mathbf{A}\in\mathcal{U}_\Delta} \{\mathbf{x}^*(\mathbf{I}-(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n})(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n})(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}\}\\
&= \theta_2(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n}).
\end{aligned}$$

Therefore, $\Gamma(\mathbf{G}) = \Gamma(\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n})$. We conclude that if $\mathbf{G}$ is an optimal solution of (3.1), then so is $\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n}$ for all $\sigma\in\mathcal{A}$. Hence, by the convexity of $\Gamma$ it follows that the convex combination $\frac{1}{N}\sum_{\sigma\in\mathcal{A}}\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n}$ is also an optimal solution. However, it can be easily verified that $\frac{1}{N}\sum_{\sigma\in\mathcal{A}}\mathbf{P}_{\sigma,m}\mathbf{G}\mathbf{P}^*_{\sigma,n} = \mathcal{C}(\mathbf{G}_0,\mathbf{G}_1,\ldots,\mathbf{G}_{N-1})$ for some matrices $\mathbf{G}_0,\mathbf{G}_1,\ldots,\mathbf{G}_{N-1}\in\mathbb{C}^{m\times n}$. Specifically, if

$$\mathbf{G} = \begin{pmatrix}
\mathbf{G}_{00} & \mathbf{G}_{01} & \cdots & \mathbf{G}_{0,N-1}\\
\mathbf{G}_{10} & \mathbf{G}_{11} & \cdots & \mathbf{G}_{1,N-1}\\
\vdots & \vdots & & \vdots\\
\mathbf{G}_{N-1,0} & \mathbf{G}_{N-1,1} & \cdots & \mathbf{G}_{N-1,N-1}
\end{pmatrix},$$

then $\mathbf{G}_k = \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{G}_{i,i+k}$, $0\le k\le N-1$. $\square$

**3.2. SDP formulation of the estimation problem.** We now use Theorem 3.1 to develop an SDP formulation of (1.4). We first consider the inner maximization problem

$$\max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2} \mathbf{x}^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}. \tag{3.2}$$

As a result of Theorem 3.1, we can assume that $\mathbf{G}$ is a BC matrix. Since $\mathbf{I}$, $\mathbf{T}$, and $\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}$ are also BC matrices, it follows that $\mathbf{H}\equiv\mathbf{T}^{-1/2}(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{T}^{-1/2}$ is a BC matrix. By the properties listed in Lemma 2.1, we can deduce that for every $0\le j\le N-1$

$$\begin{aligned}\mathbf{F}_j(\mathbf{H}) &= \mathbf{F}_j(\mathbf{T})^{-1/2}\mathbf{F}_j((\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A})))\mathbf{F}_j(\mathbf{T})^{-1/2}\\ &= \mathbf{S}_j\left(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A}))\right)^*\left(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A}))\right)\mathbf{S}_j,\end{aligned} \tag{3.3}$$

where $\mathbf{S}_j=\mathbf{F}_j(\mathbf{T})^{-1/2}$ and $\mathbf{E}_j=\mathbf{F}_j(\mathbf{G})$. Therefore, by Theorem 2.1, we have

$$\begin{aligned}&\max_{\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2,\,\boldsymbol{\Delta}\mathbf{A}\in\mathcal{U}_\Delta} \mathbf{x}^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))^*(\mathbf{I}-\mathbf{G}(\mathbf{A}+\boldsymbol{\Delta}\mathbf{A}))\mathbf{x}\\ &\quad= NL^2\max_{\boldsymbol{\Delta}\mathbf{A}\in\mathcal{U}_\Delta}\max_{0\le j\le N-1}\alpha_j(\boldsymbol{\Delta}\mathbf{A}),\end{aligned} \tag{3.4}$$

where $\alpha_j(\boldsymbol{\Delta}\mathbf{A})$ is given by

$$\lambda_{\max}\left(\mathbf{S}_j(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})))^*(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})))\mathbf{S}_j\right). \tag{3.5}$$

We can therefore express (3.4) as the solution to the problem

$$\min_{\tau} NL^2\tau \tag{3.6}$$

subject to

$$\mathbf{S}_j(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})))^*(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})))\mathbf{S}_j\preceq\tau\mathbf{I} \tag{3.7}$$

for every $\boldsymbol{\Delta}\mathbf{A}\in\mathcal{U}_\Delta$. Invoking Schur's lemma [4], we can rewrite the constraint (3.7) as

$$\begin{pmatrix}\tau\mathbf{I} & \mathbf{S}_j(\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})))^*\\ (\mathbf{I}-\mathbf{E}_j(\mathbf{F}_j(\mathbf{A})+\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})))\mathbf{S}_j & \mathbf{I}\end{pmatrix}\succeq 0,$$

which can be further written as

$$\mathbf{R}_j\succeq\mathbf{P}_j^*\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})\mathbf{Q}_j+\mathbf{Q}_j^*\mathbf{F}_j(\boldsymbol{\Delta}\mathbf{A})^*\mathbf{P}_j\quad\forall\boldsymbol{\Delta}\mathbf{A}\in\mathcal{U}_\Delta, \tag{3.8}$$

where

$$\mathbf{R}_j=\begin{pmatrix}\tau\mathbf{I} & \mathbf{S}_j(\mathbf{I}-\mathbf{E}_j\mathbf{F}_j(\mathbf{A}))^*\\ (\mathbf{I}-\mathbf{E}_j\mathbf{F}_j(\mathbf{A}))\mathbf{S}_j & \mathbf{I}\end{pmatrix},$$
$$\mathbf{P}_j=\begin{pmatrix}\mathbf{0} & \mathbf{E}_j^*\end{pmatrix},\quad \mathbf{Q}_j=\begin{pmatrix}\mathbf{S}_j & \mathbf{0}\end{pmatrix}.$$

We now exploit the following lemma, the proof of which is very similar to the proof of Lemma 2 in [8] and thus is omitted here.

LEMMA 3.1. $\mathbf{P},\mathbf{Q},\mathbf{R}$ $\mathbf{R}=\mathbf{R}^*$ $\mathcal{U}_\Delta$ (1.5)

$$\mathbf{R}\succeq\mathbf{P}^*\mathbf{F}_j(\mathbf{X})\mathbf{Q}+\mathbf{Q}^*\mathbf{F}_j(\mathbf{X})^*\mathbf{P}\qquad\mathbf{X}\in\mathcal{U}_\Delta\qquad 0\le j\le N-1$$

[illegible] $\lambda \geq 0$ [illegible]

$$\begin{pmatrix} \mathbf{R} - \lambda \mathbf{Q}^*\mathbf{Q} & -\rho \mathbf{P}^* \\ -\rho \mathbf{P} & \lambda \mathbf{I} \end{pmatrix} \succeq 0,$$

[illegible] $\rho = \sum_{j=0}^{N-1} \rho_j$

From Lemma 3.1, it follows that (3.8) is satisfied if and only if there exists $\lambda_j \geq 0$, $0 \leq j \leq N-1$, such that

$$\begin{pmatrix} \tau \mathbf{I} - \lambda_j \mathbf{F}_j(\mathbf{T})^{-1} & \mathbf{S}_j(\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A}))^* & \mathbf{0} \\ (\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A}))\mathbf{S}_j & \mathbf{I} & -\rho \mathbf{E}_j \\ \mathbf{0} & -\rho \mathbf{E}_j^* & \lambda_j \mathbf{I} \end{pmatrix} \succeq 0 \tag{3.9}$$

with $\rho = \sum_{j=0}^{N-1} \rho_j$. Summarizing the above derivations, we see that problem (1.4) reduces to

$$\min_{\tau, \lambda_j, \mathbf{G}} \left\{ \mathrm{Tr}(\mathbf{G}\mathbf{C}\mathbf{G}^*) + NL^2 \tau \right\}$$

subject to (3.9).

Since $\mathbf{C}$ and $\mathbf{G}$ are both BC matrices, the product $\mathbf{G}\mathbf{C}\mathbf{G}^*$ is also a BC matrix. Let $\mathbf{G}\mathbf{C}\mathbf{G}^* = \mathcal{C}(\mathbf{S}_0, \mathbf{S}_2, \ldots, \mathbf{S}_{N-1})$ for some $\mathbf{S}_0, \mathbf{S}_2, \ldots, \mathbf{S}_{N-1} \in \mathbb{C}^{m \times m}$. Then $\mathrm{Tr}(\mathbf{G}\mathbf{C}\mathbf{G}^*) = N \,\mathrm{Tr}(\mathbf{S}_0)$. However,

$$N\mathbf{S}_0 = N\mathbf{F}_0^{-1}(\mathbf{F}(\mathbf{G}\mathbf{C}\mathbf{G}^*)) = \sum_{j=0}^{N-1} \mathbf{F}_j(\mathbf{G}\mathbf{C}\mathbf{G}^*) = \sum_{j=0}^{N-1} \mathbf{E}_j \mathbf{F}_j(\mathbf{C}) \mathbf{E}_j^*. \tag{3.10}$$

We thus arrive at the following formulation of problem (1.4):

$$\min_{\tau, \lambda_j, \mathbf{E}_j} \left\{ NL^2 \tau + \sum_{j=0}^{N-1} \mathrm{Tr}(\mathbf{E}_j \mathbf{F}_j(\mathbf{C}) \mathbf{E}_j^*) \right\} \tag{3.11}$$

subject to

$$\begin{pmatrix} \tau \mathbf{I} - \lambda_j \mathbf{F}_j(\mathbf{T})^{-1} & \mathbf{S}_j(\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A}))^* & \mathbf{0} \\ (\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A}))\mathbf{S}_j & \mathbf{I} & -\rho \mathbf{E}_j \\ \mathbf{0} & -\rho \mathbf{E}_j^* & \lambda_j \mathbf{I} \end{pmatrix} \succeq 0, \quad 0 \leq j \leq N-1, \tag{3.12}$$

which is equivalent to

$$\min_{\tau, t_j, \mathbf{E}_j, \lambda_j} \left\{ \sum_{j=0}^{N-1} t_j + NL^2 \tau \right\} \tag{3.13}$$

subject to the LMI (3.12) and

$$\mathrm{Tr}(\mathbf{E}_j \mathbf{F}_j(\mathbf{C}) \mathbf{E}_j^*) \leq t_j, \quad 0 \leq j \leq N-1, \tag{3.14}$$

which can clearly be expressed as an LMI (see (3.15)). Thus, our problem reduces finally to an SDP.

We summarize our results in Theorem 3.2, where we present the SDP formulation for the circulant model.

Theorem 3.2 (SDP formulation). [illegible] 3.1 [illegible] (1.4) [illegible] $\mathbf{G}=\mathcal{C}(\mathbf{G}_0,\dots,\mathbf{G}_{N-1})$ [illegible]

$$\mathbf{G}_j=\frac{1}{N}\sum_{k=0}^{N-1}\omega^{-kj}\mathbf{E}_k,\quad 0\le j\le N-1.$$

[illegible] $\omega=e^{-\frac{2\pi i}{N}}$ [illegible] $\mathbf{E}_j$ $0\le j\le N-1$ [illegible]

$$\min_{\tau,\lambda_j,t_j,\mathbf{E}_j}\left\{NL^2\tau+\sum_{j=0}^{N-1}t_j\right\}$$

[illegible]

$$\begin{pmatrix} t_j & \mathbf{e}_j^* \\ \mathbf{e}_j & \mathbf{I}\end{pmatrix}\succeq 0,\quad 0\le j\le N-1, \tag{3.15}$$

$$\begin{pmatrix} \tau\mathbf{I}-\lambda_j\mathbf{F}_j(\mathbf{T})^{-1} & \mathbf{S}_j(\mathbf{I}-\mathbf{E}_j\mathbf{F}_j(\mathbf{A}))^* & \mathbf{0} \\ (\mathbf{I}-\mathbf{E}_j\mathbf{F}_j(\mathbf{A}))\mathbf{S}_j & \mathbf{I} & -\rho\mathbf{E}_j \\ \mathbf{0} & -\rho\mathbf{E}_j^* & \lambda_j\mathbf{I}\end{pmatrix}\succeq 0,$$

[illegible] $\mathbf{e}_j=\mathrm{vec}(\mathbf{E}_j\mathbf{F}_j(\mathbf{C})^{1/2})$ $\mathbf{S}_j=\mathbf{F}_j(\mathbf{T})^{-1/2}$ [illegible] $\rho=\sum_{j=0}^{N-1}\rho_j$

**4. Minimax MSE estimator for $\mathbf{T}=\mathbf{I}$.** In this section we discuss a special case of the minimax MSE estimator problem where $\mathbf{T}=\mathbf{I}$. When $\mathbf{A}$ is certain, we find an explicit expression for the optimal minimax MSE estimator. In the case of uncertain $\mathbf{A}$, we show that the SDP problem of Theorem 3.2 can be reduced to a simple convex optimization problem in $N+1$ unknowns.

**4.1. Minimax MSE estimator for $\mathbf{T}=\mathbf{I}$ with known $\mathbf{A}$.** In the case of known $\mathbf{A}$, we return to the problem of a single system $\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{\Delta y}$ with $\mathbf{x}^*\mathbf{T}\mathbf{x}\le L^2$. This problem was discussed in [8], where it was shown that the minimax MSE estimator for the case $\mathbf{T}=\mathbf{I}$ is given by $\hat{\mathbf{x}}=\mathbf{G}\mathbf{y}$ with

$$\mathbf{G}=\alpha(\mathbf{A}^*\mathbf{C}^{-1}\mathbf{A})^{-1}\mathbf{A}^*\mathbf{C}^{-1}, \tag{4.1}$$

where $\alpha=\frac{L^2}{L^2+B}$ and

$$B=\mathrm{Tr}\left((\mathbf{A}^*\mathbf{C}^{-1}\mathbf{A})^{-1}\right). \tag{4.2}$$

The estimator of (4.1) is a [illegible] proposed by Mayer and Willke [19], which is simply a scaled version of the LS estimator with an optimal choice of shrinkage factor.

Note that the dominant computation in (4.1) and (4.2) is the inversion of the $mN\times mN$ matrix $\mathbf{A}^*\mathbf{C}^{-1}\mathbf{A}$, which requires $O(m^3N^3)$ operations. This number is prohibitively large even for medium size problems. On the other hand, the calculation stemming from Theorem 4.1, which exploits the BC structure, requires the inversion of $N$ DFT components, each an $m\times m$ matrix resulting in a total of only $O(m^3N)$ operations. For example, if $N=100$, then our computation is 10000 cheaper than the direct computation.

Theorem 4.1. [illegible] $\mathbf{x}$ [illegible] $\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{\Delta y}$ [illegible] $\mathbf{A}$ [illegible] $\mathbf{\Delta y}$ [illegible] $\mathbf{C}$ [illegible]

$\min_{\mathbf{G}} \max_{\|\mathbf{x}\|^2 \le L^2} E(\|\hat{\mathbf{x}} - \mathbf{x}\|^2)$ [illegible] $\mathbf{G} = \mathcal{C}(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{N-1})$ [illegible]

$$\mathbf{G}_j = \frac{1}{N} \sum_{k=0}^{N-1} \omega^{-kj} \mathbf{E}_k, \quad 0 \le j \le N-1.$$

[illegible]

$$\mathbf{E}_j = \frac{L^2}{L^2 + B} \left(\mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{C})^{-1} \mathbf{F}_j(\mathbf{A})\right)^{-1} \mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{C})^{-1}, \quad 0 \le j \le N-1,$$

[illegible] $B = \sum_{j=0}^{N-1} \operatorname{Tr}\left((\mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{C})^{-1} \mathbf{F}_j(\mathbf{A}))^{-1}\right)$

[illegible] First, we note that $B$ of (4.2) is equal to $\sum_{i=1}^{mN} \frac{1}{\lambda_i}$, where $\lambda_1, \lambda_2, \ldots, \lambda_{mN}$ are the eigenvalues of $\mathbf{A}^* \mathbf{C}^{-1} \mathbf{A}$. From Theorem 2.1, it follows that

$$B = \sum_{j=0}^{N-1} \operatorname{Tr}\left((\mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{C})^{-1} \mathbf{F}_j(\mathbf{A}))^{-1}\right).$$

By Theorem 3.1, $\mathbf{G}$ is a BC matrix and thus is equal to $\mathcal{C}(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{N-1})$ for some $\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{N-1} \in \mathbb{C}^{m \times n}$. Using the properties listed in Lemma 2.1, we can calculate the $j$th DFT component of $\mathbf{G}$ (denoted by $\mathbf{E}_j$):

$$\begin{aligned} \mathbf{E}_j &= \mathbf{F}_j \left( \frac{L^2}{L^2 + B} (\mathbf{A}^* \mathbf{C}^{-1} \mathbf{A})^{-1} \mathbf{A}^* \mathbf{C}^{-1} \right) \\ &= \frac{L^2}{L^2 + B} (\mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{C})^{-1} \mathbf{F}_j(\mathbf{A}))^{-1} \mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{C})^{-1}. \end{aligned}$$

Applying the inverse DFT, we obtain the desired expression for $\mathbf{G}_j$, and the result follows. □

As can be expected intuitively, when $L \to \infty$, the minimax MSE estimator $\hat{\mathbf{x}} = \mathbf{G}\mathbf{y}$ of Theorem 4.1 reduces to the LS estimator. Indeed, when the norm of $\mathbf{x}$ can be made arbitrarily large, the MSE will also be arbitrarily large unless the bias is equal to zero. Therefore, in this limit, the worst-case estimation error is minimized by choosing an estimator with zero bias that minimizes the variance, which leads to the LS solution.

### 4.2. Minimax estimator for $\mathbf{T} = \mathbf{I}$, $\mathbf{C} = \sigma^2\mathbf{I}$, and unknown model matrix.

We now show that in the case where $\mathbf{T} = \mathbf{I}$ and $\mathbf{C} = \sigma^2 \mathbf{I}$, the minimax MSE estimator reduces to a simple convex optimization problem in $N+1$ unknowns.

THEOREM 4.2. [illegible] 3.1 [illegible] $\mathbf{C} = \sigma^2 \mathbf{I}$ [illegible] $0 \le j \le N-1$ [illegible] $\mathbf{F}_j(\mathbf{A}) = \mathbf{U}_j \Sigma_j \mathbf{V}_j^*$ [illegible] $\mathbf{F}_j(\mathbf{A})$ [illegible] $j$ [illegible] $\mathbf{A}$ [illegible] $\Sigma_j$ [illegible] $n \times m$ [illegible] $\sigma_{j,k} > 0$ $1 \le k \le m$ [illegible] $\mathbf{U}_j$ [illegible] $\mathbf{V}_j$ [illegible] $\min_{\mathbf{G}} \max_{\|\mathbf{x}\|^2 \le L^2, \Delta\mathbf{A} \in \mathcal{U}_\Delta} E(\|\hat{\mathbf{x}} - \mathbf{x}\|^2)$ [illegible] $\mathbf{G} = \mathcal{C}(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{N-1})$ [illegible] $\mathbf{G}_j = \mathbf{F}_j^{-1}(\mathbf{E})$ $\mathbf{E} = \mathcal{C}(\mathbf{E}_0, \ldots, \mathbf{E}_{N-1})$ [illegible]

$$\mathbf{E}_j = \mathbf{V}_j \mathbf{Z}_j \mathbf{V}_j^* (\mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{A}))^{-1/2} \mathbf{F}_j(\mathbf{A})^*, \quad 0 \le j \le N-1,$$

[illegible] $\mathbf{Z}_j$ [illegible] $m \times m$ [illegible] $z_{j,k} = f_{j,k}(\tau, \lambda_j)$ [illegible]

$$f_{j,k}(\tau, \lambda_j) = \frac{\sigma_{j,k}\lambda_j - \sqrt{\lambda_j(\tau - \lambda_j)\left(\sigma_{j,k}^2 \lambda_j - \rho^2(1 + \lambda_j - \tau)\right)}}{(\tau - \lambda_j)\rho^2 + \sigma_{j,k}^2 \lambda_j},$$

[illegible] $\rho = \sum_{j=0}^{N-1} \rho_j$, [illegible] $\lambda_0, \dots, \lambda_{N-1}$, [illegible] $\tau$ [illegible]

$$\min_{\tau, \lambda_j} \left\{ \sigma^2 \sum_{j=0}^{N-1} \sum_{k=1}^{m} f_{j,k}^2(\tau, \lambda_j) + NL^2\tau \right\}$$

[illegible]

$$\begin{aligned} \lambda_j \sigma_{j,k}^2 &\geq \rho^2(1 + \lambda_j - \tau), \quad 1 \leq k \leq m, \quad 0 \leq j \leq N-1, \\ \lambda_j &\geq 0, \quad 0 \leq j \leq N-1, \\ \tau &\geq \lambda_j, \quad 0 \leq j \leq N-1. \end{aligned}$$

[illegible] From Theorem 3.2, the optimal estimator $\mathbf{G}$ is equal to $\mathcal{C}(\mathbf{G}_0, \dots, \mathbf{G}_{N-1})$, where $\mathbf{G}_j = \frac{1}{N} \sum_{k=0}^{N-1} \omega^{-kj} \mathbf{E}_k$ and $(\mathbf{E}_j)_{j=0}^{N-1}$ is the solution to

$$\min_{\tau, \mathbf{E}_j, \lambda_j} \left\{ \sigma^2 \sum_{j=0}^{N-1} \operatorname{Tr}(\mathbf{F}_j(\mathbf{E}) \mathbf{F}_j(\mathbf{E})^*) + NL^2\tau \right\}, \tag{4.3}$$

subject to

$$\mathbf{M}_j \triangleq \begin{pmatrix} (\tau - \lambda_j)\mathbf{I} & (\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A}))^* & \mathbf{0} \\ (\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A})) & \mathbf{I} & -\rho \mathbf{E}_j \\ \mathbf{0} & -\rho \mathbf{E}_j^* & \lambda_j \mathbf{I} \end{pmatrix} \succeq 0. \tag{4.4}$$

The proof of the theorem is comprised of three parts. First, we show that the optimal solution $(\mathbf{E}_j)_{j=0}^{N-1}$ to (4.3) and (4.4) is of the form

$$\mathbf{E}_j = \mathbf{V}_j \mathbf{Z}_j \mathbf{V}_j^* \left( \mathbf{F}_j(\mathbf{A})^* \mathbf{F}_j(\mathbf{A}) \right)^{-1/2} \mathbf{F}_j(\mathbf{A})^*, \quad 0 \leq j \leq N-1, \tag{4.5}$$

for some $m \times m$ matrices $\mathbf{Z}_0, \mathbf{Z}_1, \dots, \mathbf{Z}_{N-1}$. We then show that $\mathbf{Z}_0, \mathbf{Z}_1, \dots, \mathbf{Z}_{N-1}$ can be chosen as diagonal matrices. Finally, we find the diagonal elements of $\mathbf{Z}_0, \mathbf{Z}_1, \dots, \mathbf{Z}_{N-1}$.

We begin by showing that the optimal $(\mathbf{E}_j)_{j=0}^{N-1}$ has the form (4.5). The constraint (4.4) is equivalent to $\mathbf{Q}_j \mathbf{M}_j \mathbf{Q}_j^* \succeq 0$ for any invertible $\mathbf{Q}_j$. Choosing

$$\mathbf{Q}_j = \begin{pmatrix} \mathbf{V}_j^* & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_j^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_j^* \end{pmatrix}, \quad 0 \leq j \leq N-1,$$

(4.4) becomes

$$\begin{pmatrix} (\tau - \lambda_j)\mathbf{I} & \mathbf{V}_j^* (\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A}))^* \mathbf{V}_j & \mathbf{0} \\ \mathbf{V}_j^* (\mathbf{I} - \mathbf{E}_j \mathbf{F}_j(\mathbf{A})) \mathbf{V}_j & \mathbf{I} & -\rho \mathbf{V}_j^* \mathbf{E}_j \mathbf{U}_j \\ \mathbf{0} & -\rho \mathbf{U}_j^* \mathbf{E}_j^* \mathbf{V} & \lambda_j \mathbf{I} \end{pmatrix} \succeq 0. \tag{4.6}$$

Making the change of variables

$$\mathbf{B}_j \triangleq \mathbf{V}_j^* \mathbf{E}_j \mathbf{U}_j, \tag{4.7}$$

so that

$$\mathbf{E}_j = \mathbf{V}_j \mathbf{B}_j \mathbf{U}_j^*, \tag{4.8}$$

the problem of (4.3) and (4.6) can be expressed as

$$\min_{\tau,\lambda_j,\mathbf{B}_j}\left\{\sigma^2\sum_{j=0}^{N-1}\mathrm{Tr}(\mathbf{B}_j^*\mathbf{B}_j)+NL^2\tau\right\} \tag{4.9}$$

subject to

$$\begin{pmatrix} (\tau-\lambda_j)\mathbf{I} & (\mathbf{I}-\mathbf{B}_j\Sigma_j)^* & \mathbf{0}\\ (\mathbf{I}-\mathbf{B}_j\Sigma_j) & \mathbf{I} & -\rho\mathbf{B}_j\\ \mathbf{0} & -\rho\mathbf{B}_j^* & \lambda_j \end{pmatrix}\succeq 0. \tag{4.10}$$

Let $\mathbf{B}_j=(\mathbf{Z}_j\ \mathbf{W}_j)$, where $\mathbf{Z}_j$ is the $m\times m$ matrix consisting of the first $m$ columns of $\mathbf{B}_j$, and let $\widetilde{\Sigma}_j$ denote the $m\times m$ matrix with diagonal elements $\sigma_{j,k}$, $1\le k\le m$, for every $0\le j\le N-1$. Then we can express the constraint (4.10) as

$$\mathbf{L}(\mathbf{B}_j)\triangleq\begin{pmatrix} (\tau-\lambda_j)\mathbf{I} & (\mathbf{I}-\mathbf{Z}_j\widetilde{\Sigma}_j)^* & \mathbf{0} & \mathbf{0}\\ (\mathbf{I}-\mathbf{Z}_j\widetilde{\Sigma}_j) & \mathbf{I} & -\rho\mathbf{Z}_j & -\rho\mathbf{W}_j\\ \mathbf{0} & -\rho\mathbf{Z}_j^* & \lambda_j\mathbf{I} & \mathbf{0}\\ \mathbf{0} & -\rho\mathbf{W}_j^* & \mathbf{0} & \lambda_j\mathbf{I} \end{pmatrix}\succeq 0. \tag{4.11}$$

Clearly, if (4.11) is satisfied, then

$$\mathbf{K}(\mathbf{Z}_j)\triangleq\begin{pmatrix} (\tau-\lambda_j)\mathbf{I} & (\mathbf{I}-\mathbf{Z}_j\widetilde{\Sigma}_j)^* & \mathbf{0}\\ (\mathbf{I}-\mathbf{Z}_j\widetilde{\Sigma}_j) & \mathbf{I} & -\rho\mathbf{Z}_j\\ \mathbf{0} & -\rho\mathbf{Z}_j^* & \lambda_j \end{pmatrix}\succeq 0. \tag{4.12}$$

Now let $\mathbf{B}_j=(\mathbf{Z}_j\ \mathbf{W}_j)$ be any matrix satisfying (4.11), and define $\widetilde{\mathbf{B}}_j=(\mathbf{Z}_j\ \mathbf{0})$. Then

$$\mathbf{L}(\widetilde{\mathbf{B}}_j)=\begin{pmatrix} \mathbf{K}(\mathbf{Z}_j) & \mathbf{0}\\ \mathbf{0} & \lambda_j \end{pmatrix}\succeq\mathbf{0},$$

since $\mathbf{K}(\mathbf{Z}_j)\succeq\mathbf{0}$. In addition,

$$\mathrm{Tr}(\widetilde{\mathbf{B}}_j^*\widetilde{\mathbf{B}}_j)=\mathrm{Tr}(\mathbf{Z}_j^*\mathbf{Z}_j)\le\mathrm{Tr}(\mathbf{Z}_j^*\mathbf{Z}_j)+\mathrm{Tr}(\mathbf{W}_j^*\mathbf{W}_j)=\mathrm{Tr}(\mathbf{B}_j^*\mathbf{B}_j).$$

Therefore, the optimal value of $\mathbf{B}_j$ satisfies $\mathbf{W}_j=\mathbf{0}$ for every $0\le j\le N-1$, so that the problem of (4.9) and (4.10) reduces to

$$\min_{\tau,\mathbf{Z}_j,\lambda_j}\left\{\sigma^2\sum_{j=0}^{N-1}\mathrm{Tr}(\mathbf{Z}_j^*\mathbf{Z}_j)+NL^2\tau\right\}, \tag{4.13}$$

subject to (4.12). Once we find the optimal $(\mathbf{Z}_j)_{j=0}^{N-1}$, the optimal $(\mathbf{E}_j)_{j=0}^{N-1}$ can be found from (4.8) as

$$\mathbf{E}_j=\mathbf{V}_j\mathbf{Z}_j(\mathbf{I}\ \mathbf{0})\mathbf{U}_j^*=\mathbf{V}_j\mathbf{Z}_j\mathbf{V}_j^*(\mathbf{F}_j(\mathbf{A})^*\mathbf{F}_j(\mathbf{A}))^{-1/2}\mathbf{F}_j(\mathbf{A})^*,$$

thus completing the first part of the proof.

We now show that the optimal values of $(\mathbf{Z}_j)_{j=0}^{N-1}$ can be chosen as diagonal matrices. To this end, we first note that if $(\mathbf{Z}_j)_{j=0}^{N-1}$ satisfies (4.12), then for every

$0 \le j \le N-1$

$$
\begin{aligned}
&\widetilde{\mathbf{J}}\begin{pmatrix} (\tau-\lambda_j)\mathbf{I} & (\mathbf{I}-\mathbf{Z}_j\widetilde{\Sigma}_j)^* & \mathbf{0} \\ (\mathbf{I}-\mathbf{Z}_j\widetilde{\Sigma}_j) & \mathbf{I} & -\rho\mathbf{Z}_j \\ \mathbf{0} & -\rho\mathbf{Z}_j^* & \lambda_j \end{pmatrix}\widetilde{\mathbf{J}} \\
&= \begin{pmatrix} (\tau-\lambda_j)\mathbf{I} & (\mathbf{I}-\mathbf{J}\mathbf{Z}_j\mathbf{J}\widetilde{\Sigma}_j)^* & \mathbf{0} \\ (\mathbf{I}-\mathbf{J}\mathbf{Z}_j\mathbf{J}\widetilde{\Sigma}_j) & \mathbf{I} & -\rho\mathbf{J}\mathbf{Z}_j\mathbf{J} \\ \mathbf{0} & -\rho\mathbf{J}\mathbf{Z}_j^*\mathbf{J} & \lambda_j \end{pmatrix} \succeq 0, \qquad (4.14)
\end{aligned}
$$

where $\mathbf{J}$ is any diagonal matrix with diagonal elements $\pm 1$, $\widetilde{\mathbf{J}} = \operatorname{diag}(\mathbf{J},\mathbf{J},\mathbf{J})$, and we have used the fact that diagonal matrices commute and that $\mathbf{J}^*\mathbf{J} = \mathbf{J}^2 = \mathbf{I}$. It follows from (4.14) that $\mathbf{K}(\widetilde{\mathbf{Z}}_j) \succeq \mathbf{0}$ for any $\mathbf{J}$, where $\widetilde{\mathbf{Z}}_j = \mathbf{J}\mathbf{Z}_j\mathbf{J}$. In addition, we have that $\operatorname{Tr}(\widetilde{\mathbf{Z}}_j^*\widetilde{\mathbf{Z}}_j) = \operatorname{Tr}(\mathbf{Z}_j^*\mathbf{Z}_j)$. Therefore, if $(\mathbf{Z}_j)_{j=0}^{N-1}$ is an optimal solution, then so is $(\mathbf{J}\mathbf{Z}_j\mathbf{J})_{j=0}^{N-1}$. Since our problem is convex, the set of optimal solutions is also convex [18], which implies that $(\mathbf{Z}'_j)_{j=0}^{N-1} = ((1/2^m)\sum_{\mathbf{J}}\mathbf{J}\mathbf{Z}_j\mathbf{J})_{j=0}^{N-1}$ is also a solution, where the summation is over all $2^m$ diagonal matrices $\mathbf{J}$ with diagonal elements $\pm 1$. It is easy to see that $\mathbf{Z}'_j$ is a diagonal matrix. Therefore, we have shown that there exists an optimal diagonal solution $\mathbf{Z}_j$ for every $0 \le j \le N-1$.

Denote the diagonal elements of $\mathbf{Z}_j$ by $z_{j,k}$, $1 \le k \le m$, and let $\operatorname{diag}(\alpha_1,\dots,\alpha_m)$ denote the $m \times m$ diagonal matrix with diagonal elements $\alpha_j$. By permuting the rows and the columns of the matrix $\mathbf{K}(\mathbf{Z}_j)$, it can be seen that the constraint $\mathbf{K}(\mathbf{Z}_j) \succeq \mathbf{0}$ can be written as

$$
\begin{pmatrix} \tau-\lambda_j & 1-\sigma_{j,k}z_{j,k} & 0 \\ 1-\sigma_{j,k}z_{j,k} & 1 & -\rho z_{j,k} \\ 0 & -\rho z_{j,k} & \lambda_j \end{pmatrix}, \quad 1 \le k \le m. \qquad (4.15)
$$

Thus, the problem of (4.13) and (4.12) becomes

$$
\min_{\tau, z_{j,k}, \lambda_j} \left\{ \sigma^2 \sum_{j=0}^{N-1}\sum_{i=1}^{m} z_{j,k}^2 + NL^2\tau \right\} \qquad (4.16)
$$

subject to

$$
\begin{pmatrix} \tau-\lambda_j & 1-\sigma_{j,k}z_{j,k} & 0 \\ 1-\sigma_{j,k}z_{j,k} & 1 & -\rho z_{j,k} \\ 0 & -\rho z_{j,k} & \lambda_j \end{pmatrix} \succeq 0 \qquad (4.17)
$$

for every $1 \le k \le m$, $0 \le j \le N-1$. We now show that the problem of (4.16) subject to (4.17) can be further simplified. First, we note that to satisfy (4.17) we must have that

$$
\tau \ge \max_{0 \le j \le N-1} \lambda_j.
$$

Suppose first that $\tau > \max_{0 \le j \le N-1}\lambda_j$. In this case, by Schur's lemma, (4.17) is equivalent to

$$
\begin{aligned}
&\begin{pmatrix} 1 & -\rho z_{j,k} \\ -\rho z_{j,k} & \lambda_j \end{pmatrix} - \frac{1}{\tau-\lambda_j}\begin{pmatrix} 1-\sigma_{j,k}z_{j,k} \\ 0 \end{pmatrix}\begin{pmatrix} 1-\sigma_{j,k}z_{j,k} & 0 \end{pmatrix} \\
&= \begin{pmatrix} 1-\frac{(1-\sigma_{j,k}z_{j,k})^2}{\tau-\lambda} & -\rho z_{j,k} \\ -\rho z_{j,k} & \lambda_j \end{pmatrix} \succeq 0. \qquad (4.18)
\end{aligned}
$$

Now a $2 \times 2$ matrix is positive semidefinite if and only if the diagonal elements and the determinant are nonnegative. Therefore, (4.18) is equivalent to the conditions

$$\lambda_j \geq 0, \tag{4.19}$$

$$\tau - \lambda_j \geq (1 - \sigma_{j,k} z_{j,k})^2, \tag{4.20}$$

$$\lambda_j \left(1 - \frac{(1 - \sigma_{j,k} z_{j,k})^2}{\tau - \lambda_j}\right) - \rho^2 z_{j,k}^2 \geq 0. \tag{4.21}$$

Clearly, (4.21) and (4.19) together imply (4.20). Furthermore, we can express (4.21) as

$$z_{j,k}^2 \left((\lambda_j - \tau)\rho^2 - \sigma_{j,k}^2 \lambda_j\right) + 2 z_{j,k} \sigma_{j,k} \lambda_j + \lambda_j (\tau - \lambda_j - 1) \geq 0. \tag{4.22}$$

Since the coefficient multiplying $z_{j,k}^2$ in (4.22) is negative, it follows that there exists a $z_{j,k}$ satisfying (4.22) if and only if the discriminant is nonnegative, i.e, if and only if

$$\sigma_{j,i}^2 \lambda_j + \left((\tau - \lambda_j)\rho^2 + \sigma_{j,i}^2 \lambda_j\right)(\tau - \lambda_j - 1) \geq 0.$$

Using the fact that $\tau - \lambda_j > 0$ for every $0 \leq j \leq N - 1$, the latter inequality is equivalent to

$$\lambda_j \sigma_{j,k}^2 \geq \rho^2 (1 + \lambda_j - \tau). \tag{4.23}$$

If (4.23) is satisfied, then the set of $z_{j,k}$'s satisfying (4.22) are

$$z_{j,k}^- \leq z_{j,k} \leq z_{j,k}^+,$$

where $z_{j,k}^- \leq z_{j,k}^+$ are the roots of the quadratic function in (4.22). Since we would like to choose $z_{j,k}$ to minimize (4.16), it follows that the optimal $z_{j,k}$ is

$$\begin{aligned} z_{j,k} &= f_{j,k}(\tau, \lambda_j) \\ &= \frac{\sigma_{j,k}\lambda_j - \sqrt{\lambda_j(\tau - \lambda_j)\left(\sigma_{j,k}^2 \lambda_j - \rho^2(1 + \lambda_j - \tau)\right)}}{(\tau - \lambda_j)\rho^2 + \sigma_{j,k}^2 \lambda_j}. \end{aligned} \tag{4.24}$$

Thus, if $\tau > \max_{0 \leq j \leq N-1} \lambda_j$, then the optimal value of $z_{j,k}$ is given by (4.24), where, in addition, conditions (4.23) and (4.19) must be satisfied.

Next, suppose that $\tau = \lambda_j$ for some $j$. In this case, to ensure that (4.17) is satisfied, we must have that

$$z_{j,i} = \frac{1}{\sigma_{j,k}}, \tag{4.25}$$

$$\lambda_j \geq \frac{\rho^2}{\sigma_{j,k}^2}. \tag{4.26}$$

We can immediately verify that (4.25) and (4.26) are special cases of (4.24) and (4.23) with $\tau = \lambda_j$. We therefore conclude that the optimal value of $z_{j,k}$ is given by (4.24) subject to (4.23) and (4.19). Substituting the optimal value of $z_{j,k}$ into (4.16), our problem becomes

$$\min_{\tau, \lambda_j} \left\{ \sigma^2 \sum_{j=0}^{N-1} \sum_{k=1}^{m} f_{j,k}^2(\tau, \lambda_j) + N L^2 \tau \right\} \tag{4.27}$$

subject to

$$\begin{aligned} \lambda_j \sigma_{j,k}^2 &\geq \rho^2(1+\lambda_j-\tau), \quad 1 \leq k \leq m, \quad 0 \leq j \leq N-1, \\ \lambda_j &\geq 0, \quad 0 \leq j \leq N-1, \\ \tau &\geq \lambda_j, \quad 0 \leq j \leq N-1. \end{aligned} \tag{4.28}$$

Since the problem of (4.16) subject to (4.17) is convex, and the reduced problem (4.27) subject to (4.28) is obtained by minimizing over some of the variables in (4.16), the reduced problem is also convex, completing the proof of the theorem. □

[illegible] 4.1. The line of analysis employed in Theorem 4.2 can also be carried out when $\mathbf{T} = (\mathbf{A}^*\mathbf{A})^{\alpha}$ for some real number $\alpha$. The resulting optimization problem is very similar to the one derived in Theorem 4.2. In some applications such as the image deblurring examples described in [8], choosing a negative $\alpha$ provides better results than the Euclidean weighting (i.e., $\alpha = 0$).

**5. An image deblurring example.** To illustrate the effectiveness of the minimax MSE approach, we consider an image deblurring example from the "Regularization Tools" [14].

We consider the square system

$$\mathbf{A}_{\text{true}}\mathbf{x}_{\text{true}} = \mathbf{y}_{\text{true}},$$

where $\mathbf{x}_{\text{true}} \in \mathbb{R}^{1024}$ is obtained by stacking the columns of the $32 \times 32$ image and $\mathbf{A}_{\text{true}}$ is a $1024 \times 1024$ matrix that represents an atmospheric turbulence blur originating from [13] and implemented in the function blur(n,3,0.5) from the "Regularization Tools" [14] (3 is the half bandwidth and 0.5 is the standard deviation associated with the corresponding point spread function). The image corresponding to $\mathbf{x}_{\text{true}}$ is shown at the top of Figure 1. The matrix $\mathbf{A}_{\text{true}}$ is a block Toeplitz matrix with half bandwidth 3. We note that, in fact, any matrix representing a two-dimensional convolution has a block Toeplitz structure [16].

The observed matrix $\mathbf{A}$ was generated by the function blur(n,3,0.7), and so essentially the uncertainty in the model matrix is due to lack of knowledge of the standard deviation. The observed vector was generated by adding white noise $\mathbf{y} = \mathbf{y}_{\text{true}} + \sigma\mathbf{e}$, where each component of $\mathbf{e} \in \mathbb{R}^{1024}$ was generated from a standard normal distribution.

In our experiment the standard deviation $\sigma$ was chosen to be 0.1, which results with the noisy image shown in Figure 1 (Observation). We considered several estimation methods:

- [illegible]. The LS estimator is given by $\hat{\mathbf{x}}_{\text{LS}} = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*\mathbf{y}$. As can be seen in Figure 1, the resulting image is of a poor quality.
- [illegible]. The structured TLS (STLS) solution $\hat{\mathbf{x}}_{\text{STLS}}$ to the problem is the $\mathbf{x}$-part of the solution to the optimization problem

  $$\min_{\Delta\mathbf{A},\Delta\mathbf{y},\mathbf{x}} \{\|\mathbf{\Delta A}\|^2 + \|\mathbf{\Delta y}\|^2 : (\mathbf{A}+\mathbf{\Delta A})\mathbf{x} = \mathbf{y}+\mathbf{\Delta y},\ \mathbf{\Delta A} \text{ is BC}\}.$$

  The STLS problem with BC structure can be solved by decomposing the problem into several unstructured TLS problems (for details see [1]). As can be seen from Figure 1, the STLS method generates an even worse image than $\hat{\mathbf{x}}_{\text{LS}}$. This poor performance of the STLS solution stems from the fact that the unstructured TLS solution is a deregularization [15] of the LS solution and as such is rather unstable. The STLS solution for BC systems is constructed

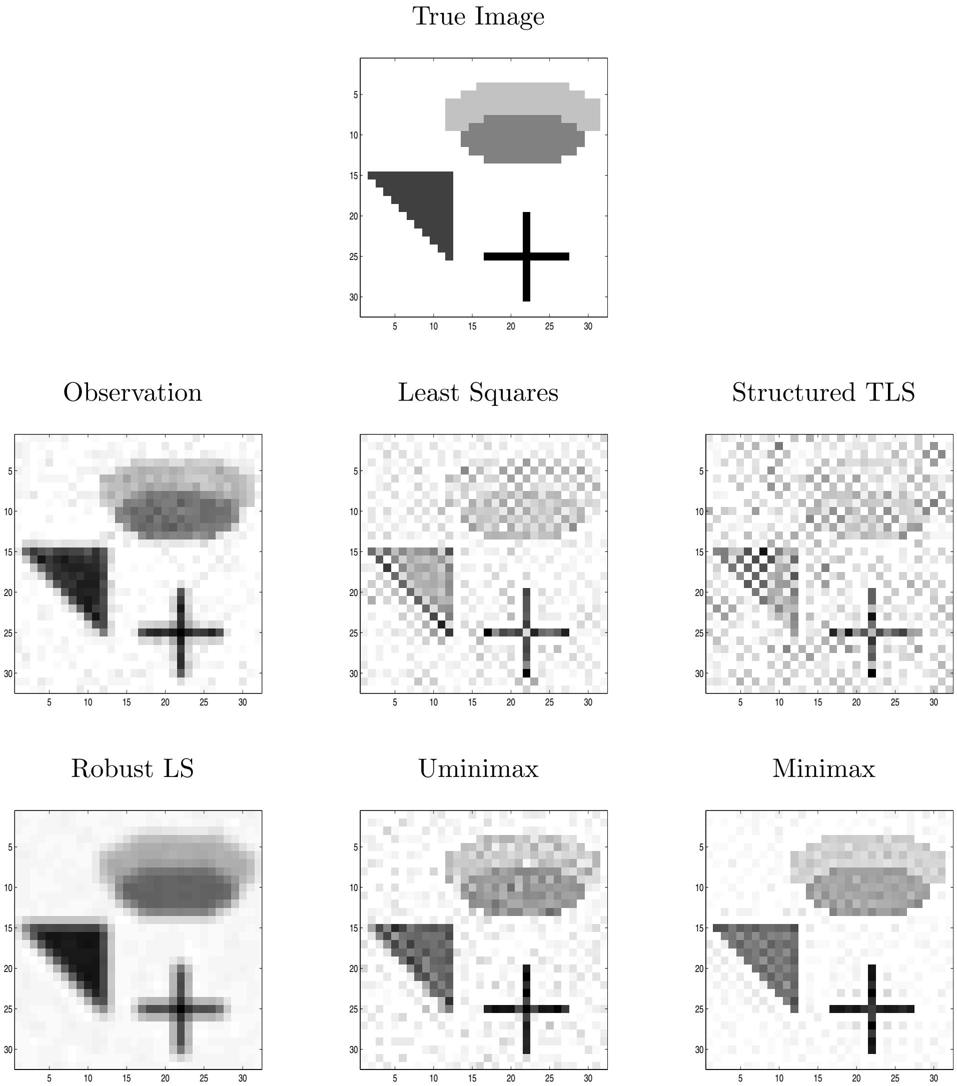


FIG. 1. *Comparison between different estimators.*

from several solutions of unstructured TLS problems and is therefore unstable as well.

- [illegible]. We also considered the RLS method defined in (1.1), where the uncertainty set $\mathcal{U}$ is given by a simple norm constraint $\mathcal{U} = \{(\boldsymbol{\Delta}\mathbf{A}, \boldsymbol{\Delta}\mathbf{y}) : \|(\boldsymbol{\Delta}\mathbf{A}, \boldsymbol{\Delta}\mathbf{y})\| \leq \rho_{\mathrm{R}}\}$ and $\rho_{\mathrm{R}}$ is chosen as $1.1 \cdot \|(\mathbf{A} - \mathbf{A}_{\mathrm{true}}, \mathbf{y} - \mathbf{y}_{\mathrm{true}})\|$. The resulting figure is quite blurred. The reason for not using a complicated set such as $\mathcal{U}_{\Delta}$ (given in (1.5)) to describe the uncertainty in $\mathbf{A}$ is that problem (1.1) appears to be intractable in this case, since the uncertainty set involves [illegible] norm constraints. Another alternative would be to use the structured RLS problem [10] and to relax the multiple norm constraints in $\mathcal{U}_{\Delta}$ into a single norm constraint. However, the generated SDP needed to be solved in our example here is too large to handle with standard software.

- *Uminimax*. Unstructured minimax is the minimax estimator for the unstructured case (see [8]). This estimator minimizes the worst-case MSE across all values of $\mathbf{x}$ satisfying $\mathbf{x}^*\mathbf{T}\mathbf{x} \leq L^2$ and perturbation matrices $\mathbf{\Delta A}$ satisfying $\|\mathbf{\Delta A}\| \leq \rho_B$. Note, however, that it ignores the special structure of $\mathbf{\Delta A}$. We have chosen the parameters $L, \rho_B$ to be 10 percent larger than their true values (for example, $L$ was chosen to be $1.1 \cdot \mathbf{x}_{\text{true}}^*\mathbf{T}\mathbf{x}_{\text{true}}$). $\mathbf{T}$ was chosen to be $(\mathbf{A}^*\mathbf{A})^{-1}$. This choice of $\mathbf{T}$ reflects the fact that components corresponding to small singular values of $\mathbf{A}^*\mathbf{A}$ should receive a smaller weight than components corresponding to large singular values. The resulting image for this method is of good quality.
- *Minimax*. Finally, we compared the above-mentioned methods with the minimax MSE estimator for BC systems developed in this paper. In implementing the Minimax estimator, we have used a BC approximation of the block Toeplitz matrix $\mathbf{A}$ as follows:

$$\begin{pmatrix}
\mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0
\end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix}
\mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{A}_{-2} & \mathbf{A}_{-1} \\
\mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{A}_{-2} \\
\mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 \\
\mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{-1} & \mathbf{A}_0 & \mathbf{A}_1 \\
\mathbf{A}_1 & \mathbf{A}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{-2} & \mathbf{A}_{-1} & \mathbf{A}_0
\end{pmatrix}.$$

  The approximation is made by adding three block matrices to the northeast and southwest corners of $\mathbf{A}$. As in the Uminimax estimator, all parameters are chosen to be 10 percent larger than their true value. It can be seen that Minimax gives even a better result than Uminimax.

We note that the Minimax estimate was not calculated by solving the SDP formulation of Theorem 3.2, since its size was too big for standard software such as SeDuMi [23]. Instead we applied a gradient projection algorithm with armijo-type line search [5] on the convex optimization formulation of Theorem 4.2. In this algorithm the dominant computational effort is the calculation of the orthogonal projection onto the polyhedral feasible set, which amounts to solving a quadratic minimization problem in 1025 variables. Since the linear system describing the feasible set is extremely sparse, the CPU time required to calculate a single projection (using SeDuMi) was a small fraction of a second. The resulting image was obtained after 10 iterations in an overall CPU time of 0.8 seconds (on a Pentium 4, 1.8 Ghz). The stopping criterion was chosen to be $|f_k - f_{k-1}| < \varepsilon$, where $\varepsilon = 10^{-3}$ and $f_j$ denotes the objective function value at the $j$th iteration. We noticed that the quality of the image does not improve if we choose a smaller value of $\varepsilon$.

As can be seen from this example, the structured minimax MSE estimator gives better results than the LS, STLS, RLS, and Uminimax estimators.

## REFERENCES

[1] A. Beck and A. Ben-Tal, *A global solution for the structured total least squares problem with block circulant matrices*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 238–255.

[2] A. Beck, A. Ben-Tal, and Y. C. Eldar, *Robust mean-squared error estimation of multiple signals in linear systems affected by model and noise uncertainties*, Math. Program., 107 (2006), pp. 155–187.

[3] Z. Ben-Haim and Y. C. Eldar, *Blind Minimax Estimation*, CCIT report 550, Electrical Engineering Department, Technion—Israel Institute of Technology, Haifa, Israel, 2005, IEEE Trans. Inform. Theory, submitted.

[4] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS/SIAM Ser. Optim. 2, SIAM, Philadelphia, 2001.

[5] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed., Athena Scientific, Belmont, MA, 1999.

[6] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed, *Parameter estimation in the presence of bounded data uncertainties*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 235–252.

[7] Y. C. Eldar, A. Ben-Tal, and A. Nemirovski, *Linear minimax regret estimation of deterministic parameters with bounded data uncertainties*, IEEE Trans. Signal Process., 52 (2004), pp. 2177–2188.

[8] Y. C. Eldar, A. Ben-Tal, and A. Nemirovski, *Robust mean-squared error estimation in the presence of model uncertainties*, IEEE Trans. Signal Process., 53 (2005), pp. 168–181.

[9] H. Gazzah, P. A. Regalia, and J. Delmas, *Asymptotic eigenvalue distribution of block Toeplitz matrices and application to blind SIMO channel identification*, IEEE Trans. Inform. Theory, 47 (2001), pp. 1243–1251.

[10] L. El Ghaoui and H. Lebret, *Robust solutions to least-squares problems with uncertain data*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1035–1064.

[11] G. H. Golub and C. F. Van Loan, *An analysis of the total least squares problem*, SIAM J. Numer. Anal., 17 (1980), pp. 883–893.

[12] R. Gray, *Toeplitz and Circulant Matrices: A Review*, Tech. report 6504-1, Information System Laboratory, Stanford University, Stanford, CA, 1977.

[13] M. Hanke and P. C. Hansen, *Regularization methods for large-scale problems*, Surveys Math. Indust., 3 (1993), pp. 253–315.

[14] P. C. Hansen, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35.

[15] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, Frontiers Appl. Math. 9, SIAM, Philadelphia, 1991.

[16] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1989.

[17] A. K. Katsaggelos, K. T. Lay, and N. P. Galatsanos, *A general framework for frequency domain multi-channel signal processing*, IEEE Trans. Image Process., 2 (1993), pp. 417–420.

[18] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley and Sons, New York, 1968.

[19] L. S. Mayer and T. A. Willke, *On biased estimation in linear models*, Technometrics, 15 (1973), pp. 497–508.

[20] H. Murakami, *Discrete wavelet transform based on cyclic convolution*, IEEE Trans. Signal Process., 52 (2004), pp. 165–174.

[21] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[22] A. H. Sayed and S. Chandrasekaran, *Parameter estimation with multiple sources and levels of uncertainties*, IEEE Trans. Signal Process., 48 (2000), pp. 680–692.

[23] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11/12 (1999), pp. 625–653.

[24] M. Sznaier, O. Camps, and M. C. Mazzaro, *Finite horizon model reduction of a class of neutrally stable systems with applications to texture synthesis and recognition*, in Proceedings of the 43rd IEEE Conference on Decision and Control, 2004.

[25] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95.

[26] M. Vollmer, M. Haardt, and J. Gotze, *Comparative study of joint-detection techniques for TD-CDMA based mobile radio systems*, IEEE Select. Areas Comm., 19 (2001), pp. 1461–1475.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 731–751

# ACCURATE COMPUTATIONS WITH TOTALLY NONNEGATIVE MATRICES*

PLAMEN KOEV†

**Abstract.** We consider the problem of performing accurate computations with rectangular ($m \times n$) totally nonnegative matrices. The matrices under consideration have the property of having a unique representation as products of nonnegative bidiagonal matrices. Given that representation, one can compute the inverse, LDU decomposition, eigenvalues, and SVD of a totally nonnegative matrix to high relative accuracy in $\mathcal{O}(\max(m^3, n^3))$ time—much more accurately than conventional algorithms that ignore that structure. The contribution of this paper is to show that the high relative accuracy is preserved by operations that preserve the total nonnegativity—taking a product, re-signed inverse (when $m = n$), converse, Schur complement, or submatrix of a totally nonnegative matrix, any of which costs at most $\mathcal{O}(\max(m^3, n^3))$. In other words, the class of totally nonnegative matrices for which we can do numerical linear algebra very accurately in $\mathcal{O}(\max(m^3, n^3))$ time (namely, those for which we have a product representation via nonnegative bidiagonals) is closed under the operations listed above.

**Key words.** high relative accuracy, totally positive matrix, totally nonnegative matrix, bidiagonal decomposition

**AMS subject classifications.** 65F15, 15A18

**DOI.** 10.1137/04061903X

## 1. Introduction.

The matrices with all minors nonnegative are called [illegible] and appear in a wide variety of applications [5, 10, 12, 14, 15, 18, 25]. They are often very ill conditioned, which means that conventional matrix algorithms such as LAPACK [1] may deliver little or no accuracy when solving totally nonnegative linear systems or computing inverses, eigenvalues, or SVDs.

Our goal is to derive algorithms for performing accurate and efficient computations with $m \times n$ totally nonnegative matrices. The types of computations we would like to perform include computing the inverse, LDU decomposition, eigenvalues, and SVD. By [illegible] we mean that each quantity must be computed [illegible]—it must have a correct sign and leading digits. By [illegible] we mean [illegible] $\mathcal{O}(\max(m^3, n^3))$ [illegible].

It turns out that the problem of performing accurate computations with totally nonnegative matrices is very much a [illegible]. If, instead of representing a matrix by its entries, we represent it as a product of nonnegative bidiagonal matrices

$$A = L^{(1)} L^{(2)} \cdots L^{(m-1)} D U^{(n-1)} U^{(n-2)} \cdots U^{(1)}; \tag{1.1}$$

then given the entries of $L^{(k)}$, $D$, and $U^{(k)}$, we can compute $A^{-1}$, the LDU decomposition, the eigenvalues, and the SVD of $A$ accurately and efficiently (see section 3).

The existence and uniqueness of the bidiagonal decomposition (1.1) is critical to the design of our algorithms. Therefore we restrict the class of totally nonnegative matrices under consideration to only those that are [illegible]. If the matrix under consideration

---

*Received by the editors November 16, 2004; accepted for publication (in revised form) by Uwe Helmke December 28, 2006; published electronically June 12, 2007. This work was supported by National Science Foundation grant DMS-0314286.
http://www.siam.org/journals/simax/29-3/61903.html

†Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 (plamen@math.mit.edu).

is square ($m = n$), the above restriction means that the matrix itself is [illegible]

We will call the matrices in the above described class TN for short.

The representation (1.1) is intrinsic [19] and immediately reveals the TN structure of an $m \times n$ TN matrix $A$. The $m \cdot n$ nontrivial nonnegative entries in the factors of (1.1) parameterize the set of all $m \times n$ TN matrices and determine the quantities that we would like to compute (the entries of $A^{-1}$, the entries of the LDU decomposition, the eigenvalues, and the SVD) accurately (section 3).

TN matrices can be obtained in a variety of ways as a result of matrix operations that preserve the total nonnegativity. The following result is well known [19, 20, 25].

PROPOSITION 1.1. [illegible] $A = [a_{ij}]$ [illegible] $AB$ [illegible] $B$ [illegible] converse $[a_{m+1-i,n+1-j}]$ [illegible] $R$ [illegible] $A$ [illegible] $r_{ii} > 0$ [illegible] $a_{11}$ [illegible] $A$ [illegible] $A$ [illegible] $A$ [illegible] re-signed inverse $[(-1)^{i+j}a_{ij}]^{-1}$ [illegible] $A$ [illegible] $A$ [illegible] $R$ [illegible]

If $A$ is a TN matrix obtained from other TN matrices by any sequence of these operations, the question becomes: Can we perform accurate matrix computations with $A$? In other words, if these other TN matrices are represented by their corresponding bidiagonal decompositions (1.1), can we accurately and efficiently compute the bidiagonal decomposition of $A$?

Our main contribution in this paper is to answer this question affirmatively. In section 5 we present accurate and efficient algorithms that perform these computations. These algorithms prove the following theorem.

THEOREM 1.2. [illegible] (1.1) [illegible] $m = n$ [illegible] 1.1 [illegible]

For example, we could take the product of the Hilbert matrix and the Pascal matrix, compute a Schur complement, take a submatrix of its converse, and then compute the SVD of the resulting matrix highly accurately, all in $\mathcal{O}(\max(m^3, n^3))$ time. In contrast, on examples similar to this one, the conventional algorithms may fail to compute even the largest singular value accurately (see section 7).

As an application of Theorem 1.2, in section 6 we derive a new algorithm for computing the bidiagonal decomposition of a TN generalized Vandermonde matrix based on removing appropriate columns of an ordinary Vandermonde matrix. This is a major improvement over previous such algorithms in [8, 34].

In the design of our algorithms we take the following approach.

First, we identify the source of large relative errors in conventional matrix algorithms. Relative accuracy in these algorithms is lost due to [illegible] in the subtraction of approximate same-sign quantities. Conversely, the relative accuracy is preserved in multiplication, division, addition, and taking of square roots.

Second, we perform any and all transformations listed in Proposition 1.1 as a combination of the following [illegible] (EETs):

EET1: Subtracting a multiple of a row (column) from the next in order to create a zero in such a way that the transformed matrix is still TN;

EET2: Adding a multiple of a row (column) to the previous one;

EET3: Adding a multiple of a row (column) to the next one;

EET4: Scaling by a positive diagonal matrix.

Each of these EETs preserves the total nonnegativity [19].

Third, instead of applying an EET directly on a TN matrix $A$, we carry it out [illegible] by transforming the entries of its bidiagonal decomposition, and arrange the computations in such a way that subtractions are not required. Thus the accuracy is preserved.

This paper is organized as follows. In section 2 we review the bidiagonal decompositions of TN matrices. In section 3 we review algorithms for accurate computations with TN matrices, given their bidiagonal decompositions. In section 4 we review algorithms from [27] for performing EET1 and EET2 and present new algorithms for performing EET3 and EET4. In section 5 we present algorithms for computing accurate bidiagonal decompositions of derivative TN matrices, obtained as described in Proposition 1.1. We present our new algorithm for computing the bidiagonal decomposition of a generalized Vandermonde matrix in section 6. In section 7 we present numerical results demonstrating the accuracy of our algorithms. We draw conclusions and present open problems in section 8.

[illegible] Throughout this paper we use MATLAB [32] notation for vectors and submatrices.

**2. Bidiagonal decompositions of TN matrices.** The TN matrices possess a very elegant structure, which is not revealed by their entries. Additionally, small relative perturbations in the entries of a TN matrix $A$ can cause enormous relative perturbations in the small eigenvalues, singular values, and entries of $A^{-1}$ [27, section 1]. Thus the matrix entries are ill suited as parameters in numerical computations with TN matrices.

Instead, following [27], we choose to represent a TN matrix as a product of nonnegative bidiagonal matrices. This representation arises naturally in the process of [illegible], which we now review, following [19] (see also [35]).

In the process of Neville elimination a matrix is reduced to upper triangular form using only [illegible] rows. A zero is introduced in position $(m, 1)$ by subtracting a multiple $b_{m1} = a_{m1}/a_{m-1,1}$ of row $m-1$ from row $m$. Subtracting the multiple $b_{m-1,1} = a_{m-1,1}/a_{m-2,1}$ of row $m-2$ from row $m-1$ creates a zero in position $(m-1, 1)$, and so on. The total nonnegativity is preserved during Neville elimination [19], and therefore all multipliers $b_{ij}$ are nonnegative.

This yields the decomposition

$$A = \left( \prod_{k=1}^{m-1} \prod_{j=m-k+1}^{m} E_j(b_{j,k+j-m}) \right) \cdot U,$$

where $U$ is $m \times n$ upper triangular and

$$E_j(x) \equiv \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & x & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}$$

is $m \times m$ and differs from the identity only in the $(j, j-1)$ entry.

Applying the same process to $A^T$, we obtain the decomposition

$$A = \left( \prod_{k=1}^{m-1} \prod_{j=m-k+1}^{m} E_j(b_{j,k+j-m}) \right) \cdot D \cdot \left( \prod_{1}^{k=n-1} \prod_{n-k+1}^{j=n} E_j^T(b_{k+j-n,j}) \right), \tag{2.1}$$

where $D$ is a diagonal $m \times n$ matrix and $E_j^T$ are $n \times n$. In the notation of (2.1) and throughout this paper, $\prod_1^{k=n-1}$ indicates that the product is taken for $k$ from $n-1$ down to 1. Although somewhat nonstandard, this notation allows us to preserve the symmetry in (2.1).

The matrices

$$L^{(k)} \equiv \prod_{j=m-k+1}^{m} E_j(b_{j,k+j-m}) = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & b_{k,m-k+1} & 1 & & & \\ & & b_{k+1,m-k+2} & 1 & & \\ & & & \ddots & \ddots & \\ & & & & b_{m,m-1} & 1 \end{bmatrix}$$

and

$$U^{(k)} \equiv \prod_{n-k+1}^{j=n} E_j^T(b_{k+j-n,j}) = \begin{bmatrix} 1 & & & & & \\ & \ddots & b_{n-k+1,k} & & & \\ & & 1 & b_{n-k+2,k+1} & & \\ & & & 1 & \ddots & \\ & & & & \ddots & b_{n-1,n} \\ & & & & & 1 \end{bmatrix}$$

are $m \times m$ lower- and $n \times n$ upper bidiagonal, respectively. The decomposition (2.1) now becomes

$$A = L^{(1)} \cdots L^{(n-1)} \cdot D \cdot U^{(n-1)} \cdots U^{(1)}.$$

We denote the off-diagonal entries in $L^{(k)}$ and $U^{(k)}$ as

$$l_i^{(k)} \equiv L_{i+1,i}^{(k)} = b_{i+1,k+i+1-m} \quad \text{and} \quad u_i^{(k)} \equiv U_{i,i+1}^{(k)} = b_{k+i+1-n,i+1}. \tag{2.2}$$

We will use either $l_i^{(k)}$ or $b_{i+1,k+i+1-m}$ to denote the nontrivial entries of $L^{(k)}$ (and similarly with $U^{(k)}$). In different contexts one notation may be more convenient than the other, so we will keep (2.2) in mind when switching back and forth.

We now present the fundamental structure theorem for TN matrices.

THEOREM 2.1 (Gasca and Peña [19]). [illegible] $m \times n$ [illegible] $A$ [illegible]

$$A = L^{(1)} \cdots L^{(m-1)} \cdot D \cdot U^{(n-1)} \cdots U^{(1)}, \tag{2.3}$$

[illegible] $D$ [illegible] $m \times n$ [illegible] $d_i, i = 1, 2, \ldots, \min(n, m)$. $L^{(k)}$ [illegible] $U^{(k)}$ [illegible] $m \times m$ [illegible] $n \times n$ [illegible]

1. $d_i > 0$ for $i = 1, 2, \ldots, \min(m, n)$.
2. $l_i^{(k)} = 0,\ i < m - k$; $u_i^{(k)} = 0,\ i < n - k$; and $l_i^{(k)} = u_i^{(k)} = 0,\ i > m + n - k$.
3. $l_i^{(k)} \geq 0,\ m - k \leq i \leq m + n - k$ and $u_i^{(k)} \geq 0,\ n - k \leq i \leq m + n - k$.
4. $l_i^{(k)} = 0$ implies $l_{i+1}^{(k-1)} = 0$. $u_i^{(k)} = 0$ implies $u_{i+1}^{(k-1)} = 0$

We will refer to Theorem 2.1 to verify whether a particular decomposition of a TN matrix $A$ as a product of bidiagonal matrices is in fact its unique bidiagonal decomposition.

Following [27], we denote the bidiagonal decomposition (2.3) of a TN matrix $A$ as $\mathcal{BD}(A)$. We store the nontrivial entries of $\mathcal{BD}(A)$ compactly in an $m \times n$ array, which we also refer to as $\mathcal{BD}(A)$:

$$(\mathcal{BD}(A))_{ij} = \begin{cases} l_{i-1}^{(n-i+j)}, & i > j, \\ u_{j-1}^{(n-j+i)}, & i < j, \\ d_i, & i = j. \end{cases}$$

The $(i, j)$th entry in $\mathcal{BD}(A)$ equals the multiplier $(b_{ij})$ used to set the $(i, j)$th entry in $A$ to zero (when $i \neq j$), or the $i$th entry on the diagonal of $D$ (when $i = j$).

For example,

$$\begin{bmatrix} 2 & 6 \\ 8 & 29 \\ 48 & 209 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 6 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 4 & 1 & \\ & 7 & 1 \end{bmatrix} \begin{bmatrix} 2 & \\ & 5 \\ & \end{bmatrix} \begin{bmatrix} 1 & 3 \\ & 1 \end{bmatrix}$$

is stored as

$$\mathcal{BD}\left( \begin{bmatrix} 2 & 6 \\ 8 & 29 \\ 48 & 209 \end{bmatrix} \right) = \left\{ \begin{matrix} 2 & 3 \\ 4 & 5 \\ 6 & 7 \end{matrix} \right\}.$$

This notation is convenient since we can formally transpose $\mathcal{BD}(A)$ to obtain $\mathcal{BD}(A^T) = (\mathcal{BD}(A))^T$ [27, section 4].

In the language of the $m \times n$ array $B = \mathcal{BD}(A)$, conditions 1–4 in Theorem 2.1 are equivalent to the following:

1. $b_{ii} > 0, i = 1, 2, \ldots, \min(m, n)$;
2. $b_{ij} = 0$, unless $1 \leq i \leq m$ and $1 \leq j \leq n$;
3. $b_{ij} \geq 0,\ i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$;
4. $b_{ij} = 0$ implies $b_{i+1,j} = 0$ if $i < j$, and $b_{i,j+1} = 0$ if $i > j$.

If the TN matrix $A$ is also totally positive (i.e., if all its minors are positive), then the entries $b_{ij}, i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$, are products and quotients of minors of $A$ [27, section 3], [16]:

$$b_{ij} = \frac{\det A(i-j+1 : i, 1 : j)}{\det A(i-j+1 : i-1, 1 : j-1)} \cdot \frac{\det A(i-j : i-2, 1 : j-1)}{\det A(i-j : i-1, 1 : j)}, \quad i > j,$$

$$b_{ii} = \frac{\det A(1 : i, 1 : i)}{\det A(1 : i-1, 1 : i-1)}; \tag{2.4}$$

$$b_{ji} = \frac{\det A(1 : j, i-j+1 : i)}{\det A(1 : j-1, i-j+1 : i-1)} \cdot \frac{\det A(1 : j-1, i-j : i-2)}{\det A(1 : j, i-j : i-1)}, \quad i > j.$$

One can use the formulas (2.4) to compute explicit formulas for the bidiagonal decompositions of Vandermonde [10, 23, 33], [27, section 3], Cauchy [4], [27, section 3], Cauchy–Vandermonde [29, 30, 31], generalized Vandermonde [8], and Bernstein–Vandermonde [28] matrices.

Neville elimination is just one of eight analogous, but slightly different methods to eliminate a TN matrix $A$ using only adjacent rows and columns [19, section 4]. Each method yields a decomposition of $A$ as a product of nonnegative bidiagonal matrices with analogous but different nonzero patterns. In section 5.1 we show how to obtain an accurate $\mathcal{BD}(A)$ (and then perform accurate computations with $A$) starting with any decomposition of $A$ as a product of nonnegative bidiagonal matrices. Therefore the particular choice of elimination pattern in Neville elimination and the resulting nonzero pattern in the factors of the decomposition (2.3) do not result in any loss of generality.

## 3. Performing accurate matrix computations given $\mathcal{BD}(A)$.

The entries of $\mathcal{BD}(A)$ determine accurately the entries of the inverse, the entries of the LDU decomposition, and the values of any minor, eigenvalue, or singular value. Furthermore, given $\mathcal{BD}(A)$, many matrix computations with $A$ can be performed accurately and efficiently. We review these results below.

### 3.1. Computing the inverse.

If $A$ is a square $n \times n$ TN matrix, we can compute its inverse accurately by inverting (2.1):

$$A^{-1} = \left( \prod_{i=1}^{n-1} \prod_{j=n-i+1}^{n} E_j^T(-b_{i+j-n,j}) \right) \cdot D^{-1} \cdot \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} E_j(-b_{j,i+j-n}) \right). \tag{3.1}$$

Using (3.1) and the Cauchy–Binet identity [13, Vol. 1, p. 9], we conclude that each entry of $A^{-1}$ is a linear function in each entry $b_{ij}$ of $\mathcal{BD}(A)$ with either nonnegative or nonpositive coefficients. Therefore small relative perturbations in the $b_{ij}$ cause small relative perturbations in any entry of $A^{-1}$. In other words, $\mathcal{BD}(A)$ determines every entry of $A^{-1}$ accurately.

We can form $A^{-1}$ by multiplying out (3.1) in $\mathcal{O}(n^3)$ time. Each entry of $A^{-1}$ will be computed accurately, since the multiplication (3.1) involves no subtractive cancellation. (All matrices in (3.1), their partial products, and $A^{-1}$ have checkerboard sign patterns.)

### 3.2. Solving $Ax = b$.

We can use (3.1) to compute the solution to $Ax = b$ in $\mathcal{O}(n^2)$ time by multiplying out the expression

$$x = A^{-1}b = \left( \prod_{i=1}^{n-1} \prod_{j=n-i+1}^{n} E_j^T(-b_{i+j-n,j}) \right) D^{-1} \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} E_j(-b_{j,i+j-n}) \right) b \tag{3.2}$$

right-to-left. The computed solution $\hat{x}$ has a small componentwise relative backward error [4]; i.e., a matrix $\hat{A}$ exists such that $\hat{A}\hat{x} = b$ and $|A - \hat{A}| \le \mathcal{O}(\epsilon)|A|$, where the inequality is meant componentwise.

If $b$ has alternating sign pattern (i.e., sign $b_i = (-1)^i$ or sign $b_i = (-1)^{i-1}$), then (3.2) involves no subtractive cancellation, and each component of $x$ is computed accurately [23].

This approach for solving $Ax = b$ is the basis of the so-called Björck–Pereyra-type methods for solving structured TN linear systems. Derived originally for Vandermonde linear systems [3], these methods received deserved attention because of their remarkable accuracy[1] [22]. Generalizations were later developed for Cauchy [4],

---

[1] In the scope of Newton interpolation with positive and increasing nodes (i.e., the conditions under which the corresponding Vandermonde matrix is TN), the accuracy observation dates back to 1963 and was made by Kahan and Farkas [24].

Cauchy–Vandermonde [29, 30, 31], generalized Vandermonde [8], and Bernstein–Vandermonde [28] matrices. Each of these methods is either explicitly or implicitly based on a decomposition of the corresponding $A^{-1}$ as a product of simple bidiagonal matrices analogous to (3.1).

**3.3. Computing a minor.** The value of any minor of a TN matrix $A$ is determined accurately by $\mathcal{BD}(A)$ [6, section 9]. It can be computed accurately and efficiently given $\mathcal{BD}(A)$—see section 5.8.

**3.4. Computing the LDU decomposition.** Let $A$ be a square TN nonsingular $n \times n$ matrix. Define

$$L \equiv L^{(1)} \cdots L^{(n-1)} \quad \text{and} \quad U \equiv U^{(n-1)} \cdots U^{(1)}. \tag{3.3}$$

Now (2.3) implies that $A = LDU$ is the LDU decomposition of $A$. The Cauchy–Binet identity and (3.3) imply that $\mathcal{BD}(A)$ determines each entry of $L$, $D$, and $U$ accurately. Multiplying out (3.3) involves no subtractions and yields every entry of $L$ and $U$ accurately. The decompositions $\mathcal{BD}(L)$ and $\mathcal{BD}(U)$ are given by (3.3).

**3.5. Computing the eigenvalues and the SVD.** In [27, section 7] we proved that $\mathcal{BD}(A)$ accurately determines the eigenvalues and the SVD of a TN matrix $A$. In the same paper we presented algorithms for computing the eigenvalues and the SVD of $A$ accurately and efficiently, given $\mathcal{BD}(A)$. These algorithms implicitly reduce both the eigenvalue and SVD problems to the bidiagonal SVD problem using only EETs. The resulting bidiagonal SVD problem is then solved accurately using known means [7, 11].

## 4. Performing EETs accurately.

Let the TN matrix $C$ be obtained from the $m \times n$ TN matrix $A$ by applying an EET to $A$. In this section we show how, given $\mathcal{BD}(A)$, the decomposition $\mathcal{BD}(C)$ can be computed without performing any subtractions.

In [27, section 4.1] we showed that EET1 is equivalent to simply setting an entry of $\mathcal{BD}(A)$ to zero; EET2 involved some "bulge chasing" in $\mathcal{BD}(A)$ [27, section 4.2] and cost at most $6(m+2)$ operations.

Next, we show how to perform EET3 and EET4 accurately.

**4.1. Adding a multiple of a row to the next one.** Let $A$ be TN and $C$ be obtained from $A$ by adding a multiple of row $i-1$ of $A$ to row $i$:

$$C = E_i(x)A, \quad x > 0.$$

In this section we show how to accurately compute $\mathcal{BD}(C)$, given $x$ and $\mathcal{BD}(A)$.

The following lemma shows how to compute the bidiagonal decomposition of the product of two bidiagonal matrices. It is the main building block of Algorithm 4.2 later in this section.

Lemma 4.1. [illegible] $B$ [illegible] $C$ [illegible] $n \times n$ [illegible] $b_i \geq 0$ [illegible] $c_i \geq 0$ $i = 1, 2, \ldots, n-1$ [illegible] $b_i = 0$ [illegible] $c_{i-1} = 0$ [illegible] $B'$ [illegible] $C'$ [illegible] $b_i' \geq 0$ [illegible] $c_i' \geq 0$ $i = 1, 2, \ldots, n-1$ [illegible] $B'C' = BC$ [illegible] $b_1' = 0$ [illegible] $b_i'$ [illegible] $c_i'$ [illegible] $4n$ [illegible]

*Proof.* We compare the entries on both sides of $B'C' = BC$,

$$\begin{bmatrix} 1 & & & & \\ 0 & 1 & & & \\ & b_2' & 1 & & \\ & & \ddots & \ddots & \\ & & & b_{n-1}' & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ c_1' & 1 & & & \\ & c_2' & 1 & & \\ & & \ddots & \ddots & \\ & & & c_{n-1}' & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ b_1 & 1 & & & \\ & b_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & b_{n-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ c_1 & 1 & & & \\ & c_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & c_{n-1} & 1 \end{bmatrix}, \tag{4.1}$$

to obtain $c_1' = b_1 + c_1$,

$$\begin{aligned} b_i' &= \frac{b_i c_{i-1}}{c_{i-1}'}, \\ c_i' &= b_i + c_i - b_i', \end{aligned} \tag{4.2}$$

for $i = 2, 3, \ldots, \min\{j | b_j = 0\}$, and $b_i' = b_i, c_i' = c_i$ otherwise. The subtraction in (4.2) can be eliminated by introducing auxiliary variables $d_i \equiv b_i - b_i'$. Then $d_1 = b_1 - b_1' = b_1$ and

$$\begin{aligned} d_i &= b_i - b_i' \\ &= b_i - \frac{b_i c_{i-1}}{c_{i-1}'} \\ &= \frac{b_i}{c_{i-1}'}(c_{i-1}' - c_{i-1}) \\ &= \frac{b_i}{c_{i-1}'}(b_{i-1} - b_{i-1}') \\ &= \frac{b_i d_{i-1}}{c_{i-1}'}. \end{aligned} \tag{4.3}$$

The subtraction-free (and therefore accurate) version of (4.2) is

$$\begin{aligned} b_i' &= \frac{b_i c_{i-1}}{c_{i-1}'}, \\ d_i &= \frac{b_i d_{i-1}}{c_{i-1}'}, \\ c_i' &= c_i + d_i. \end{aligned}$$

This computation clearly costs not more than $4n$ arithmetic operations. Since $c_i' = 0$ implies $b_{i+1}' = 0$, the product $B'C'$ is $\mathcal{BD}(BC)$. ☐

We implement the procedure from Lemma 4.1 in Algorithm 4.1 below. We overwrite $d_i$ by $d_{i+1}$, and the arrays $b$ and $c$ by $b'$ and $c'$, respectively. The quantity $e = b_{i+1}/c_i'$ is computed only once and used to update both $b_{i+1}$ and $d_{i+1}$, thus saving one division.

ALGORITHM 4.1. [illegible] 4.1 [illegible] $i$ [illegible] (4.4)[illegible]

```
function $[b, c, i]$ = dqd2$(b, c)$
$t = c_1$
$c_1 = b_1 + c_1$
$d = b_1$
$b_1 = 0$
$i = 1$
while $(i <$ length$(b))$ and $(b_{i+1} > 0)$
    $e = b_{i+1}/c_i$
    $d = ed$
    $b_{i+1} = et$
    $t = c_{i+1}$
    $c_{i+1} = c_{i+1} + d$
    $i = i + 1$
end
```

Note: The only way the product $BC$ differs from $\mathcal{BD}(BC)$ is in that $b_1 \neq 0$. The purpose of Algorithm 4.1 is to make $b_1$ zero without changing the product $BC$. No other zeros are introduced in $B$ or $C$, and no nonzeros are introduced in $B$. At most one nonzero may be introduced in $C$. Algorithm 4.1 returns the index $i$ where this may have happened ($c_i = 0$ on input, $c_i > 0$ on output). Although such an introduction of a nonzero in $C$ causes no problems in the scope of Lemma 4.1, it may require additional work and bulge chasing in Algorithm 4.2 below, which uses Algorithm 4.1 as an intermediate step.

THEOREM 4.2. [illegible] $A$ [illegible] $m \times n$ [illegible] $x > 0$[illegible] $\mathcal{BD}(A)$ [illegible] $\mathcal{BD}(E_i(x)A)$ [illegible] $4m$[illegible]

[illegible]. Let $\mathcal{BD}(A)$ be given as in (2.3),

$$A = L^{(1)}L^{(2)}\cdots L^{(m-1)}DU^{(n-1)}U^{(n-2)}\cdots U^{(1)},$$

and let $L \equiv L^{(1)}L^{(2)}\cdots L^{(m-1)}$. The TN matrix $E_i(x)L$ is TN unit lower triangular. The idea is to compute the decomposition $\mathcal{BD}(E_i(x)\cdot L)$:

$$E_i(x)L = \mathcal{L}^{(1)}\mathcal{L}^{(2)}\cdots\mathcal{L}^{(m-1)}.$$

Then $\mathcal{BD}(E_i(x)A)$ is

$$E_i(x)A = \mathcal{L}^{(1)}\mathcal{L}^{(2)}\cdots\mathcal{L}^{(m-1)}DU^{(n-1)}U^{(n-2)}\cdots U^{(1)}.$$

We use Lemma 4.1 and Algorithm 4.1 to "chase the bulge" $E_i(x)$:

$$\begin{aligned} E_i(x)L &= E_i(x)L^{(1)}L^{(2)}\cdots L^{(m-1)} \\ &= \mathcal{L}^{(1)}E_{i_1}(x_1)L^{(2)}\cdots L^{(m-1)} \\ &= \mathcal{L}^{(1)}\mathcal{L}^{(2)}E_{i_2}(x_2)\cdots L^{(m-1)} \\ &= \dots \\ &= \mathcal{L}^{(1)}\mathcal{L}^{(2)}\cdots\mathcal{L}^{(m-1)}. \end{aligned}$$

We start with $k = 1$ and repeat the following process. We apply Algorithm 4.1 to the trailing principal submatrices of $E_i(x)$ and $L^{(k)}$ consisting of rows and columns $i$ though $n$. The only nonzero in $E_i(x)$ disappears, and we obtain a new matrix $\bar{L}^{(k)} = E_i(x)L^{(k)}$.

If one of these three condition holds:

1. $k = m - 1$, or
2. no nonzeros were introduced in $\bar{L}^{(k)}$ that were not in $L^{(k)}$, or
3. a nonzero $\bar{l}_j^{(k)}$ was introduced in $\bar{L}^{(k)}$, but $l_{j-1}^{(k+1)} \neq 0$,

then we set $\mathcal{L}^{(k)} \equiv \bar{L}^{(k)}$; the "bulge chasing" is thus over, and we are done.

Otherwise (a nonzero $\bar{l}_j^{(k)}$ was introduced in $\bar{L}^{(k)}$, and $l_{j-1}^{(k+1)} = 0$, $k < m - 1$), we have $\bar{L}^{(k)} = \mathcal{L}^{(k)} \cdot E_j(\bar{l}_j^{(k)})$, where $\mathcal{L}^{(k)}$ has the same nonzero pattern as $L^{(k)}$. We set $i = j$, $x = l_j^{(k)}$, increase $k$ by one, and repeat the same process.

The computation of $\mathcal{BD}(E_i(x)A)$ is subtraction-free. At most $2n - 3$ entries in $\mathcal{BD}(A)$ are changed at not more than two arithmetic operations per entry (see Algorithm 4.1). The total cost therefore does not exceed $4n$. □

The following algorithm implements the procedure from Theorem 4.2.

ALGORITHM 4.2. [illegible] $A$ [illegible] $m \times n$ [illegible] $B = \mathcal{BD}(A)$ [illegible] $\mathcal{BD}(E_i(x)A)$ [illegible] $4n$ [illegible] $b_{jl} = 0$ [illegible] $j \notin \{1, 2, \ldots, m\}$ [illegible] $l \notin \{1, 2, \ldots, n\}$

```
function $B =$ TNAddToNext$(B, x, i)$
$[m, n] =$ size$(B)$
$z = 0$
$b_{i0} = x$
while $(z < \min(i-1, n))$ and $(b_{i-1,z} = 0)$
    for $j = 1 : m - i + 1$
        $[c_j, d_j] = b_{j+i+1, z+j-1:z+j}$
    end
    $[c, d, q] =$ dqd2$(c, d)$
    for $j = 1 : m - i + 1$
        $b_{j+i+1, z+j-1:z+j} = [c_j, d_j]$
    end
    $i = i + q - 1$
    $z = z + q$
end
```

**4.2. Multiplication by a diagonal matrix.** The product of a diagonal matrix $F = \text{diag}(f_1, \ldots, f_m)$, $f_i > 0, i = 1, 2, \ldots, m$, and an $m \times n$ TN matrix $A$ is TN. We now show how to compute $\mathcal{BD}(FA)$, given $F$ and $\mathcal{BD}(A)$.

We propagate $F$ through the factors $L^{(k)}$ in $\mathcal{BD}(A)$ using

$$\begin{bmatrix} f_1 & & & \\ & f_2 & & \\ & & \ddots & \\ & & & f_m \end{bmatrix} \begin{bmatrix} 1 & & & \\ c_1 & 1 & & \\ & \ddots & \ddots & \\ & & c_{m-1} & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ b_1 & 1 & & \\ & \ddots & \ddots & \\ & & b_{m-1} & 1 \end{bmatrix} \begin{bmatrix} f_1 & & & \\ & f_2 & & \\ & & \ddots & \\ & & & f_m \end{bmatrix},$$

where $b_i = c_i f_{i+1} / f_i$, $i = 1, 2, \ldots, m - 1$.

Algorithm 4.3. [illegible] $B = BD(A)$ [illegible] $(f_1, f_2, \ldots, f_m)$ [illegible] $\mathcal{BD}(\operatorname{diag}(f_1, f_2, \ldots, f_m) \cdot A)$ [illegible] $2mn$ [illegible]

```
function $B =$ TNDiagonalScale$(f, B)$
$[m, n] =$ size$(B)$
$b_{11} = b_{11} f_1$
for $i = 2 : m$
    if $i \le n$
        $b_{ii} = b_{ii} f_i$
    end
    $b_{i,1:\min(i-1,n)} = b_{i,1:\min(i-1,n)} \cdot f_i / f_{i-1}$
end
```

## 5. The bidiagonal decomposition of derivative TN matrices.

Let $A$ be a TN matrix obtained from other TN matrices using one of the operations listed in Proposition 1.1 that preserve the total nonnegativity.

In this section we present accurate and efficient subtraction-free algorithms for computing $\mathcal{BD}(A)$, given the corresponding bidiagonal decompositions of the input TN matrices.

MATLAB implementation of all algorithms for performing accurate computations with TN matrices presented in this paper and [27] are available online from [26].

### 5.1. A product of EETs.

Let the TN matrix $A$ be given as

$$A = F^{(1)} F^{(2)} \cdots F^{(k)},$$

where $F^{(i)}$ represents an EET; namely, it equals either $E_j(x)$, $E_j^T(x)$, or a positive diagonal matrix. Then $\mathcal{BD}(A)$ can be accurately accumulated using Algorithms 4.2, 4.3, as well as Proposition 4.1 and Algorithm 4.1 from [27].

We will use this approach throughout this section. Say we want to compute $\mathcal{BD}(A)$, where the TN matrix $A$ is obtained from other TN matrices using operations that preserve the total nonnegativity. We will represent $A$ as a product of EETs, which we will then accumulate.

Since any nonnegative bidiagonal matrix is a product of EETs, [illegible] representation $A$ as a product of nonnegative bidiagonal matrices is a good starting point for performing accurate computations with $A$. Given any such representation, we can accumulate $\mathcal{BD}(A)$ without loss of accuracy.

### 5.2. The product of TN matrices.

Let $F$ and $C$ be $m \times n$ and $n \times p$ TN matrices such that $m \le n$ or $n \ge p$. Their product $FC$ is a TN matrix. If $B = \mathcal{BD}(C)$, then from (2.1) we have

$$C = \left( \prod_{i=1}^{n-1} \prod_{j=n-i+1}^{n} E_j(b_{j,i+j-n}) \right) \cdot D \cdot \left( \prod_{1}^{i=p-1} \prod_{p-i+1}^{j=p} E_j^T(b_{i+j-p,j}) \right).$$

Therefore, forming the product $FC$ is equivalent to applying a number of EETs to $F$.

Algorithm 5.1 (product). [illegible] $F$ [illegible] $C$ [illegible] $m \times n$ [illegible] $n \times p$ [illegible] $m \le n$ [illegible] $n \ge p$ [illegible] $A = \mathcal{BD}(F)$ [illegible] $B = \mathcal{BD}(C)$ [illegible] $\mathcal{BD}(FC)$ [illegible] $\mathcal{O}(mnp)$ [illegible]

```
function $A =$ TNProduct$(A, B)$
$[m, n] =$ size$(A)$
$p =$ size$(B, 2)$
for $i = 1 : n-1$
    for $j = n-i+1 : \min(n, n+p-i)$
        $A =$ TNAddToPrevious$(A, b_{j,i+j-n}, 1, j)$
    end
end
$A = A(:, 1 : \min(n, p))$
$A =$ TNDiagonalScale$(\mathrm{diag}(B), A^T)^T$
for $i = p-1 : -1 : 1$
    for $j = p : -1 : p-i+1$
        $A =$ TNAddToNext$(A^T, b_{i+j-p,j}, j)^T$
    end
end
```

The function `TNAddToPrevious`$(A, x, 1, i)$ "adds" a multiple $x$ of column $i$ to column $i-1$ and costs at most $6(m+2)$ [27, Algorithm 4.1].

**5.3. The re-signed inverse.** Let $A$ be an $n \times n$ TN matrix, and let $J$ be a diagonal matrix of alternating 1's and $-1$'s ($J_{ii} = (-1)^{i-1}$, $i = 1, 2, \ldots, n$). The [illegible] of $A$,

$$A^* \equiv \left[(-1)^{i+j} a_{ij}\right]^{-1} = (JAJ)^{-1} = JA^{-1}J,$$

is also TN [14]. Using $J^2 = I$, $(E_i(x))^{-1} = E_i(-x)$, $JE_i(-x)J = E_i(x)$, and (2.1),

$$\begin{aligned} A^* &= J \cdot \left( \prod_{i=1}^{n-1} \prod_{j=n-i+1}^{n} E_j^T(-b_{i+j-n,j}) \right) \cdot D^{-1} \cdot \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} E_j(-b_{j,i+j-n}) \right) \cdot J \\ &= \left( \prod_{i=1}^{n-1} \prod_{j=n-i+1}^{n} JE_j^T(-b_{i+j-n,j})J \right) \cdot JD^{-1}J \cdot \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} JE_j(-b_{j,i+j-n})J \right) \\ &= \left( \prod_{i=1}^{n-1} \prod_{j=n-i+1}^{n} E_j^T(b_{i+j-n,j}) \right) \cdot D^{-1} \cdot \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} E_j(b_{j,i+j-n}) \right). \end{aligned}$$

ALGORITHM 5.2 (re-signed inverse). [illegible] $A$ [illegible] $n \times n$ [illegible] $B = \mathcal{BD}(A)$ [illegible] $C = \mathcal{BD}(A^*)$ [illegible] $\mathcal{O}(n^3)$ [illegible]

```
function $C =$ TNRSInverse$(B)$
$n =$ size$(B, 1)$
$C = I$
for $i = 1 : n-1$
    for $j = n-i+1 : n$
        $C =$ TNAddToNext$(C, b_{j,i+j-n}, j)$
    end
end
$C =$ TNDiagonalScale$((1/b_{11}, \ldots, 1/b_{nn}), C)$
for $i = n-1 : -1 : 1$
    for $j = n : -1 : n-i+1$
        $C =$ TNAddToPrevious$(C^T, b_{i+j-n,j}, 1, j)^T$
    end
end
```

**5.4. The converse.** If the $m \times n$ matrix $A = [a_{ij}]_{i,j=1}^{m,n}$ is TN, then so is its [illegible] [19]

$$A^{\#} \equiv [a_{m+1-i,n+1-j}]_{i,j=1}^{m,n}.$$

Let $B = \mathcal{BD}(A)$, and let $Y_k \equiv [\delta_{k+1-i,j}]_{i,j=1}^{k}$ be the [illegible] of size $k$. Using $Y_k^2 = I$ and $E_i^{\#}(x) = Y_k E_i(x) Y_k = E_{k+2-i}^T(x)$, we obtain

$$\begin{aligned}
A^{\#} &= Y_m A Y_n \\
&= Y_m \left( \prod_{i=1}^{m-1} \prod_{j=m-i+1}^{m} E_j(b_{j,i+j-m}) \right) D \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} E_j^T(b_{i+j-n,j}) \right) Y_n \\
&= \left( \prod_{i=1}^{m-1} \prod_{j=m-i+1}^{m} Y_m E_j(b_{j,i+j-m}) Y_m \right) Y_m D Y_n \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} Y_n E_j^T(b_{i+j-n,j}) Y_n \right) \\
&= \left( \prod_{i=1}^{m-1} \prod_{j=m-i+1}^{m} E_{m+2-j}^T(b_{j,i+j-m}) \right) D^{\#} \left( \prod_{1}^{i=n-1} \prod_{n-i+1}^{j=n} E_{n+2-j}(b_{i+j-n,j}) \right),
\end{aligned} \tag{5.1}$$

where $D^{\#} = Y_m D Y_n$ is an $m \times n$ diagonal matrix, $D_{ii}^{\#} = b_{k+1-i,k+1-i}$, $k = \min(m,n)$, $i = 1, 2, \ldots, k$. We compute $\mathcal{BD}(A^{\#})$ as the bidiagonal decomposition of the product of all EETs in (5.1).

ALGORITHM 5.3 (converse). [illegible] $B = \mathcal{BD}(A)$ [illegible] $m \times n$ [illegible] $A$ [illegible] $\mathcal{BD}(A^{\#})$ [illegible] $\mathcal{O}(mn^2)$ [illegible]

```
function $C$ = TNConverse($B$)
$[m, n]$ = size($B$)
$C$ = eye($m, n$)
for $i = 1 : m-1$
    for $j = m-i+1 : m$
        $C$ = TNAddToNext($A^T, b_{j,i+j-m}, m+2-j)^T$
    end
end
$e$ = diag($B$)
$C$ = TNDiagonalScale($e(\min(m,n) : -1 : 1), C^T)^T$
for $i = n-1 : -1 : 1$
    for $j = n : -1 : n-i+1$
        $A$ = TNAddToPrevious($A, b_{i+j-n,j}, 1, n+2-j$)
    end
end
```

**5.5. QR decomposition.** Let $A$ be TN, and let $A = QR$ be its QR decomposition such that $R$ has a positive diagonal. Then $R$ is TN and can be obtained by applying Givens rotations to $A$. Each Givens rotation preserves the TN structure of $A$ and equals the product of three EETs [27, section 4.3].

ALGORITHM 5.4 (QR decomposition). [illegible] $A$ [illegible] $m \times n$ [illegible] $A = QR$ [illegible] $A$ [illegible] $r_{ii} > 0$, $i = 1, 2, \ldots, \min(m,n)$ [illegible] $B = \mathcal{BD}(A)$ [illegible] $\mathcal{BD}(R)$ [illegible] $\mathcal{O}(mn^2)$ [illegible]

```
function $B = $ TNQR($B$)
$[m, n] = $ size($B$)
for $i = 1 : n$
    for $j = m : -1 : i+1$
        $x = b_{ji}$
        $b_{ji} = 0$
        $c = \sqrt{1 + x^2}$
        $B = ($TNAddToPrevious$(B^T, x/c, c, j))^T$
    end
end
```

**5.6. QR iteration.** Gladwell showed in [20] that if $A$ is TN and symmetric, then one step of $QR$ iteration without pivoting (provided $R$ has a positive diagonal) preserves the TN structure. We will now show how to compute the result of this iteration accurately using algorithms we already have.

Let $A$ be TN and symmetric, and let $A = LDU = QR$ be its $LDU$ and $QR$ decompositions, respectively, with $R$ having a positive diagonal.[2] Let $Q = LD_1U_1$ be the $LDU$ decomposition of $Q$ ($Q$ and $A$ share the $L$ factor).

Let $F = RQ$ be the result of one step of QR iteration performed on $A$. Then $F = RQ = RLD_1U_1$. Since $F$ is symmetric, it suffices to compute the lower bidiagonal factors and the diagonal factor of $\mathcal{BD}(F)$. Since $U_1$ is unit upper triangular, it thus suffices to compute $\mathcal{BD}(RLD_1)$. Since the factors are TN, this task is easy. We first use `TNQR` to obtain $\mathcal{BD}(R)$ and then `TNProduct` to obtain $\mathcal{BD}(RLD_1)$. We obtain $D_1$ by comparing the diagonals of the upper triangular matrices $DU = D_1U_1R$.

**5.7. The Schur complement.** Let $A$ be an $m \times n$ TN matrix, and let $A'$ be obtained from $A$ after one step of Gaussian elimination. We have $A' = KA$, where

$$
\begin{aligned}
K &= \begin{bmatrix} 1 & & & & \\ -\frac{a_{21}}{a_{11}} & 1 & & & \\ -\frac{a_{32}}{a_{11}} & & 1 & & \\ \vdots & & & \ddots & \\ -\frac{a_{m1}}{a_{11}} & & & & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & \frac{a_{m1}}{a_{m-1,1}} & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & \frac{a_{m-1,1}}{a_{m-2,1}} & 1 & \\ & & & & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{a_{31}}{a_{21}} & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \\
&\quad \times \begin{bmatrix} 1 & & & & \\ -\frac{a_{21}}{a_{11}} & 1 & & & \\ & -\frac{a_{31}}{a_{21}} & 1 & & \\ & & \ddots & \ddots & \\ & & & -\frac{a_{m1}}{a_{m-1,1}} & 1 \end{bmatrix} \\
&= \prod_{3}^{i=m} E_i(b_{i1}) \times \prod_{i=2}^{m} E_i(-b_{i1})
\end{aligned}
$$

---

[2] Technically, since $A$ is symmetric, $U = L^T$, but this is unimportant here.

is a product of EETs. Forming the product

$$\left(\prod_{i=2}^{m} E_i(-b_{i1})\right) \cdot A$$

is equivalent to using adjacent rows to zero out the first column of $A$. It is therefore equivalent to simply setting $b_{i1} = 0,\ i = 2, 3, \ldots, m$ [27, section 4.1].

The multiplications by $E_i(b_{i1}),\ i = m, \ldots, 3$, are performed using Algorithm 4.2.

ALGORITHM 5.5 (Schur complement). [illegible] $A$ [illegible] $m \times n$ [illegible] $A'$ [illegible] $A$ [illegible] $B = \mathcal{BD}(A)$ [illegible] $\mathcal{BD}(A')$ [illegible] $\mathcal{O}(mn)$

```
function $B$ = TNSchurComplement($B$)
$m$ = size($B, 1$)
$c = B(:, 1)$
$B(2 : m, 1) = 0$
for $i = 3 : m$
    $B$ = TNAddToNext($B, c_i, i$)
end
```

**5.8. A submatrix.** Any submatrix $C$ of a TN matrix $A$ is TN. In this section we show how to compute $\mathcal{BD}(C)$, given $\mathcal{BD}(A)$. It suffices to describe how to compute $\mathcal{BD}(C)$ when $C$ is obtained by removing row $i$ from $A$. We assume that $C$ is TN.

Consider first the case $i = 1$, i.e., $C$ is obtained by removing the first row of $A$:

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}, \quad C = \begin{bmatrix} a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix}.$$

Let

$$\mathcal{BD}(A) = \left\{ \begin{matrix} b_{11} & b_{12} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \ldots & b_{mn} \end{matrix} \right\}, \quad \mathcal{BD}(C) = \left\{ \begin{matrix} f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{matrix} \right\}.$$

Let $A'$ be obtained from $A$ by using adjacent columns to zero out the first row of $A$ above the main diagonal:

$$A' = A \cdot E_n^T(-b_{1n}) \cdot E_{n-1}^T(-b_{1,n-1}) \cdots E_2^T(-b_{12}). \tag{5.2}$$

Then

$$A' = \begin{bmatrix} a'_{11} & 0 & \ldots & 0 \\ a'_{21} & a'_{22} & \ldots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \ldots & a'_{mn} \end{bmatrix} \quad \text{and} \quad \mathcal{BD}(A') = \left\{ \begin{matrix} b_{11} & 0 & \ldots & 0 \\ b_{21} & b_{22} & \ldots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \ldots & b_{mn} \end{matrix} \right\}.$$

Let $C'$ be obtained by removing the first row of $A'$. From (5.2) we have

$$C' = C \cdot E_n^T(-b_{1n}) \cdot E_{n-1}^T(-b_{1,n-1}) \cdots E_2^T(-b_{12}),$$

which implies

$$C = C' \cdot E_2^T(b_{12}) \cdot E_3^T(b_{13}) \cdots E_n^T(b_{1n}). \tag{5.3}$$

Therefore, it suffices to obtain $\mathcal{BD}(C')$. (Then we will use Algorithm 4.2 to obtain $\mathcal{BD}(C)$ using (5.3).)

Let

$$C' = \begin{bmatrix} a'_{21} & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a'_{m1} & a'_{m2} & \cdots & a'_{mn} \end{bmatrix} \quad \text{and} \quad \mathcal{BD}(C') = \left\{ \begin{array}{cccc} f'_{21} & f'_{22} & \cdots & f'_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f'_{m1} & f'_{m2} & \cdots & f'_{mn} \end{array} \right\}.$$

Consider the process of Neville elimination applied to $A'$ and $C'$ to eliminate the entries $a'_{jk}$, $j \neq k, k+1$, and reduce $A'$ and $C'$ to lower and upper bidiagonal matrices $\bar{A}$ and $\bar{C}$, respectively. The same multipliers will be used in this elimination:

$$f'_{jk} = b_{jk} \quad \text{for} \quad j \neq k, k+1.$$

The matrix

$$\bar{C} = \begin{bmatrix} f'_{21} & & & \\ & f'_{32} & & \\ & & f'_{43} & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} 1 & f'_{22} & & \\ & 1 & f'_{33} & \\ & & 1 & \\ & & & \ddots \end{bmatrix} \tag{5.4}$$

is obtained by removing the first row of

$$\bar{A} = \begin{bmatrix} 1 & & & \\ b_{21} & 1 & & \\ & b_{32} & 1 & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} b_{11} & & & \\ & b_{22} & & \\ & & b_{33} & \\ & & & \ddots \end{bmatrix}. \tag{5.5}$$

By comparing entries in (5.4) and (5.5), we obtain

$$f'_{i+1,i} = b_{i+1,i} b_{ii} \quad \text{and} \quad f'_{i+1,i+1} = \frac{b_{i+1,i+1}}{b_{i+1,i} b_{ii}}.$$

We have obtained the entire $\mathcal{BD}(C')$.

Now consider the general case—we remove the $i$th row of $A$ to obtain $C$. In the process of Neville elimination, the same multipliers will be used to eliminate the first row of $C$ that were used to eliminate the first row of $A$.

We emulate the Neville elimination of the first [illegible] of $C$ by eliminating the first column of $A$ in a slightly different order. We use adjacent rows to eliminate the entries of the first column of $A$ with the exception of $a_{i+1,1}$. We use row $i-1$ to eliminate $a_{i+1,1}$—the exact same row that would be used to eliminate $a_{i+1,1}$ in $C$ using [illegible] rows.

This Gaussian-type elimination of rows $i$ and $i+1$ in $A$ can be handled in the same way as in section 5.7. We represent the elimination of rows $i$ and $i+1$ as a sequence of three EETs:

$$\begin{bmatrix} 1 & & \\ -\frac{a_i}{a_{i-1}} & 1 & \\ -\frac{a_{i+1}}{a_{i-1}} & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ -b_{i1} & 1 & \\ -b_{i+1,1}b_{i1} & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & \\ & 1 & \\ & b_{i+1,1} & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ -b_{i1} & 1 & \\ & -b_{i+1,1} & 1 \end{bmatrix}$$

$$= E_{i+1}(b_{i+1,1})E_i(-b_{i1})E_{i+1}(-b_{i+1,1}).$$

We then proceed by induction. We eliminate the second row and the second column of $A$ and so on until we have eliminated the first $i$ rows and the first $i$ columns of $A$. Now we are in familiar territory—we need to remove the first row of the trailing submatrix $A(i:m, i:n)$.

ALGORITHM 5.6 (submatrix). [illegible] $A$ [illegible] $m \times n$ [illegible] $C$ [illegible] $i$ [illegible] $A$ [illegible] $\mathcal{BD}(A)$ [illegible] $\mathcal{BD}(C)$ [illegible] $\mathcal{O}(n^2)$ [illegible].

```
function $B$ = TNSubmatrix($B, i$)
$[m, n]$ = size($B$)
if $i < m$
    for $j = 1 : \min(i-1, n)$
        $B(j+1:m, j+1:n)$ = TNAddToNext($B(j+1:m, j+1:n), b_{i+1,j}, i-j+1$)
        $b_{i+1,j} = b_{i+1,j}b_{ij}$
    end
    for $j = \min(n, m) + (m > n) : -1 : i+1$
        $b_{j,j-1} = b_{j,j-1}b_{j-1,j-1}$
        if $j \le n$
            $b_{jj} = b_{jj}/b_{j,j-1}$
        end
    end
    for $j = i+1 : n$
        $B(i+1:m, i:n)$ = TNAddToNext($B(i+1:m, i:n)^T, b_{ij}, j-i+1)^T$
    end
end
Remove the $i$th row of $B$
```

**6. Generalized Vandermonde matrices.** In this section we describe how to easily, accurately, and efficiently compute the bidiagonal decomposition of a TN generalized Vandermonde matrix

$$G \equiv \left[x_i^{j-1+\lambda_{n-j+1}}\right]_{i,j=1}^n$$

with $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n \ge 0$, $\lambda_i \in \mathbb{Z}, i = 1, 2, \ldots, n$. The matrix $G$ is well known to be TN when $0 < x_1 < x_2 < \cdots < x_n$ [14, p. 76]. The nodes $x_i$ and the [illegible] $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ of $|\lambda| = \lambda_1 + \lambda_2 + \cdots + \lambda_n$ are typical parameters used to describe generalized Vandermonde matrices. When $\lambda = (0, \ldots, 0)$, $G$ reduces to the ordinary Vandermonde matrix $V \equiv \left[x_i^{j-1}\right]_{i,j=1}^n$.

There have been a couple of attempts at deriving accurate algorithms for this class of matrices, and both have shortcomings.

In 1977 Van de Vel [34] proposed a subtraction-free algorithm for the LDU decomposition of $G$. While accuracy was clearly guaranteed, efficiency was not. Recently, motivated by this result and some theoretical arguments [6, section 9.1(2)], Demmel and the current author presented an accurate algorithm for computing $\mathcal{BD}(G)$ [8, 9]. While this algorithm is accurate and efficient (its complexity is bounded by $\mathcal{O}(n^2|\lambda|^{2+\rho}\lambda_1^{3+\rho})$, where $\rho$ is tiny [8, (3.9)]), it requires extended precision arithmetic when computing the Schur function in the intermediate steps [9]. This is a drawback.

With the results of this paper we are finally able to put this issue to rest by presenting a new very simple algorithm for computing $\mathcal{BD}(G)$, which is accurate and efficient—it costs only $\mathcal{O}(n^2\lambda_1)$ (and is thus much more efficient than the algorithm in [8]) and does not require the use of extended precision arithmetic. Once we have $\mathcal{BD}(G)$, we can clearly perform virtually all linear algebra with $G$ at a modest $\mathcal{O}(n^3)$ additional cost.

Our idea is very simple: Start with the rectangular (ordinary) TN Vandermonde matrix

$$F = \left[x_i^{j-1}\right]_{i=1,j=1}^{n,n-1+\lambda_1}.$$

The decomposition $\mathcal{BD}(F)$ is readily available in $\mathcal{O}(n(n+\lambda_1))$ time using the formulas in [27, section 3, (3.6)]. We can then use Algorithm 5.6 to remove the appropriate $\lambda_1$ columns of $F$ (at the cost of $\mathcal{O}(n^2)$ per column) to obtain $\mathcal{BD}(G)$. The total cost is nicely bounded by $\mathcal{O}(n^2\lambda_1)$.

## 7. Numerical experiments.

The algorithms presented in this paper can be used to perform a variety of accurate computations with TN matrices. We performed many tests to confirm their correctness and accuracy. In this section we present two numerical examples which incorporate several techniques for computing with TN matrices and demonstrate the accuracy and significance of our new algorithms.

For our experiments we selected two well-known notoriously ill-conditioned TN matrices—Hilbert and Pascal:

$$H = \left[\frac{1}{i+j-1}\right]_{i,j=1}^{m,n} \quad \text{and} \quad P = \left[\binom{i+j}{i}\right]_{i,j=1}^{n,p}.$$

We selected $m = 20, n = 30$, and $p = 20$, yielding fairly ill-conditioned $H$ and $P$: $\kappa(H) = 3.3 \cdot 10^{25}$ and $\kappa(P) = 1.2 \cdot 10^{20}$. Both experiments involved the product $T = HP$, which was also severely ill-conditioned: $\kappa(T) = 6 \cdot 10^{45}$.

In our first experiment, we computed the singular values of $T = HP$ using the MATLAB implementations of our accurate algorithms[3]

(7.1) `TNSingularValues(TNProduct(TNCauchyBD(1:m,0:n-1),ones(n,p)))`

and also via the conventional MATLAB call

(7.2) `svd(H*P).`

For verification, we formed $H$ and $P$, computed their product $T$, and computed $T$'s singular values in 70-digit decimal floating point arithmetic using the MATLAB function `vpa`. Since $\kappa(T) = 6 \cdot 10^{45}$, `vpa` returned the singular values of $T$ with at least 16 correct decimal digits in each. The results of `vpa` agreed to at least 14 digits with the ones computed using (7.1), confirming the accuracy of our algorithms.

---

[3] `TNSingularValues` is Algorithm 6.1 from [27], `TNProduct` is Algorithm 5.1 (see section 5); `TNCauchyBD` computes the bidiagonal decomposition of $H$ accurately using the formulas from [27, section 3]; the entries of $\mathcal{BD}(P)$ are all ones, i.e., $\mathcal{BD}(P)$ equals `ones(n,p)`.

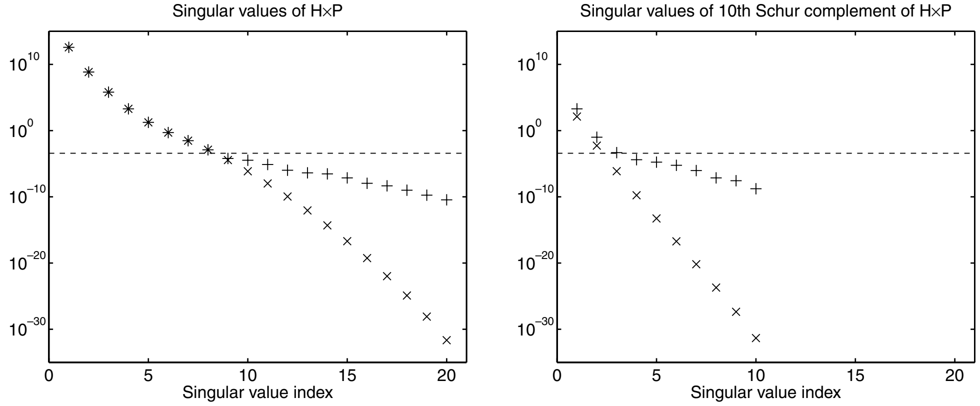


FIG. 7.1. *The singular values of the product $T$ of the $20 \times 30$ Hilbert matrix $H$ and the $30 \times 20$ Pascal matrix $P$ (left plot) and the* 10*th Schur complement of $T$ (right plot); "$\times$" = new, accurate algorithms, "$+$" = conventional. The dashed line represents the roundoff threshold, $\|T\| \cdot \varepsilon$.*

In contrast, the conventional singular value algorithms (7.2) in double precision [2] binary floating point arithmetic computed only the largest ones ($\sigma_i > \sigma_1\varepsilon = \|T\|\varepsilon$, where $\varepsilon \approx 10^{-16}$ is the machine precision) with any relative accuracy at all.

The results of this experiment are plotted in Figure 7.1, left.

In our second experiment, we computed the singular values of the 10th Schur complement of $T$ using the same three methods—our new algorithms (in particular, `TNSchurComplement`, Algorithm 5.5) and a conventional MATLAB call, and finally verified the results in extended precision arithmetic. As expected, our new algorithms computed all singular values of the 10th Schur complement of $T$ accurately, while the conventional MATLAB call failed to compute even a single singular value accurately (Figure 7.1, right).

Although this experiment is somewhat artificially contrived, it shows that very simple TN-preserving operations can result in a situation where the conventional matrix algorithms fail to deliver any accuracy at all.

## 8. Conclusions and open problems.

Using the intrinsic representation of TN matrices as a products of bidiagonal matrices allows for accurate computations with these matrices. The cost is similar to that of the conventional algorithms, but the computations are performed to high [illegible] accuracy, as opposed to the high [illegible] accuracy of the conventional algorithms.

The singular (square) totally nonnegative matrices may not have a bidiagonal decomposition, or it may not be unique. Designing new algorithms (or adapting the ones in this paper) to perform accurate computations with these matrices is still an open problem.

The problem of finding algorithms for computing accurate eigenvectors of TN matrices is also open. In particular, such algorithms should guarantee the intrinsic properties of the eigenvector matrix—the $j$th computed eigenvector should have $j-1$ changes of sign in its entries, and the eigenvector matrix should have an LU decomposition such that $L$ and $U^{-1}$ are TN [14, 17].

The caveat in our algorithms is that every TN matrix must be represented by its bidiagonal decomposition. While every TN matrix intrinsically possesses such a decomposition, and for many classes of structured matrices this decomposition is very easy to obtain accurately (see section 2), there are important TN matrices for which we know of no accurate and efficient way to compute their bidiagonal decompositions. Two such examples are the following:

- the TN generalized Vandermonde matrix $\left[x_i^{y_j}\right]_{i,j=1}^n$, where $0 < x_1 < \cdots < x_n$, $0 < y_1 < y_2 < \cdots < y_n$, and at least one $y_i$ is not an integer;
- the TN matrices appearing in the study of the hypergeometric function of a matrix argument [21].

**Acknowledgments.** I am deeply indebted to James Demmel for suggesting this problem and for providing motivation and insights during the preparation of this work. I also thank Froilán Dopico and Juan Manuel Peña for the careful reading of the manuscript and for pointing out inconsistencies in its earlier versions.

## REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide, Third Edition*, Software Environ. Tools 9, SIAM, Philadelphia, 1999.

[2] ANSI/IEEE, *IEEE Standard for Binary Floating Point Arithmetic*, New York, Std 754-1985 ed., 1985.

[3] Å. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893–903.

[4] T. Boros, T. Kailath, and V. Olshevsky, *A fast parallel Björck-Pereyra-type algorithm for solving Cauchy linear equations*, Linear Algebra Appl., 302/303 (1999), pp. 265–293.

[5] F. Brenti, *Combinatorics and total positivity*, J. Combin. Theory Ser. A, 71 (1995), pp. 175–218.

[6] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.

[7] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 873–912.

[8] J. Demmel and P. Koev, *The accurate and efficient solution of a totally positive generalized Vandermonde linear system*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 142–152.

[9] J. Demmel and P. Koev, *Accurate and efficient evaluation of Schur and Jack functions*, Math. Comp., 75 (2006), pp. 223–239.

[10] S. M. Fallat, *Bidiagonal factorizations of totally nonnegative matrices*, Amer. Math. Monthly, 108 (2001), pp. 697–712.

[11] K. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[12] S. Fomin and A. Zelevinsky, *Total positivity: Tests and parametrizations*, Math. Intelligencer, 22 (2000), pp. 23–33.

[13] F. Gantmacher, *The Theory of Matrices*, AMS Chelsea, Providence, RI, 1998.

[14] F. Gantmacher and M. Krein, *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems*, revised ed., AMS Chelsea, Providence, RI, 2002.

[15] M. Gasca and C. A. Micchelli, eds., *Total Positivity and Its Applications*, Math. Appl. 359, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[16] M. Gasca and J. M. Peña, *Total positivity and Neville elimination*, Linear Algebra Appl., 165 (1992), pp. 25–44.

[17] M. Gasca and J. M. Peña, *Total positivity, QR factorization, and Neville elimination*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1132–1140.

[18] M. Gasca and J. M. Peña, *Corner cutting algorithms and totally positive matrices*, in Curves and Surfaces in Geometric Design (Chamonix-Mont-Blanc, 1993), A. K. Peters, Wellesley, MA, 1994, pp. 177–184.

[19] M. Gasca and J. M. Peña, *On factorizations of totally positive matrices*, in Total Positivity and Its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996, pp. 109–130.

[20] G. M. L. Gladwell, *Total positivity and the QR algorithm*, Linear Algebra Appl., 271 (1998), pp. 257–272.

[21] K. I. Gross and D. S. P. Richards, *Total positivity, spherical series, and hypergeometric functions of matrix argument*, J. Approx. Theory, 59 (1989), pp. 224–246.

[22] N. J. Higham, *Error analysis of the Björck-Pereyra algorithms for solving Vandermonde systems*, Numer. Math., 50 (1987), pp. 613–632.

[23] N. J. Higham, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 23–41.

[24] W. Kahan and I. Farkas, *Algorithms* 167-169, Comm. ACM, 6 (1963), pp. 164–165. See also the *Certification*, Comm. ACM, 6(9):523.

[25] S. Karlin, *Total Positivity. Vol.* I, Stanford University Press, Stanford, CA, 1968.

[26] P. Koev, *Algorithms for totally nonnegative matrices*, http://www-math.mit.edu/˜plamen.

[27] P. Koev, *Accurate eigenvalues and SVDs of totally nonnegative matrices*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 1–23.

[28] A. Marco and J.-J. Martinez, *A fast and accurate algorithm for solving Bernstein–Vandermonde linear systems*, Linear Algebra Appl., 422 (2007), pp. 616–628.

[29] J. J. Martínez and J. M. Peña, *Factorizations of Cauchy-Vandermonde matrices*, Linear Algebra Appl., 284 (1998), pp. 229–237.

[30] J. J. Martínez and J. M. Peña, *Fast algorithms of Björck-Pereyra type for solving Cauchy-Vandermonde linear systems*, Appl. Numer. Math., 26 (1998), pp. 343–352.

[31] J. J. Martínez and J. M. Peña, *Factorizations of Cauchy-Vandermonde matrices with one multiple pole*, in Recent Research on Pure and Applied Algebra, Nova Science Publishers, Hauppauge, NY, 2003, pp. 85–95.

[32] The MathWorks, Inc., *MATLAB Reference Guide*, Natick, MA, 1992.

[33] H. Oruç and G. M. Phillips, *Explicit factorization of the Vandermonde matrix*, Linear Algebra Appl., 315 (2000), pp. 113–123.

[34] H. Van de Vel, *Numerical treatment of a generalized Vandermonde system of equations*, Linear Algebra Appl., 17 (1977), pp. 149–179.

[35] A. M. Whitney, *A reduction theorem for totally positive matrices*, J. Anal. Math., 2 (1952), pp. 88–92.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 752–773

# SYMMETRIC INDEFINITE PRECONDITIONERS FOR SADDLE POINT PROBLEMS WITH APPLICATIONS TO PDE-CONSTRAINED OPTIMIZATION PROBLEMS*

JOACHIM SCHÖBERL[†] AND WALTER ZULEHNER[‡]

**Abstract.** We consider large scale sparse linear systems in saddle point form. A natural property of such indefinite 2-by-2 block systems is the positivity of the (1,1) block on the kernel of the (2,1) block. Many solution methods, however, require that the positivity of the (1,1) block is satisfied everywhere. To enforce the positivity everywhere, an augmented Lagrangian approach is usually chosen. However, the adjustment of the involved parameters is a critical issue. We will present a different approach that is not based on such an explicit augmentation technique. For the considered class of symmetric and indefinite preconditioners, assumptions are presented that lead to symmetric and positive definite problems with respect to a particular scalar product. Therefore, conjugate gradient acceleration can be used. An important class of applications are optimal control problems. It is typical for such problems that the cost functional contains an extra regularization parameter. For control problems with elliptic state equations and distributed control, a special indefinite preconditioner for the discretized problem is constructed, which leads to convergence rates of the preconditioned conjugate gradient method that are not only independent of the mesh size but also independent of the regularization parameter. Numerical experiments are presented for illustrating the theoretical results.

**Key words.** saddle point problems, indefinite preconditioners, KKT systems, conjugate gradient methods, PDE-constrained optimization problems, optimal control problems

**AMS subject classifications.** 65F10, 15A12, 49M15

**DOI.** 10.1137/060660977

## 1. Introduction.

In this paper we consider large scale sparse linear systems of equations in saddle point form

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1.1}$$

where $A$ is a real, symmetric, and positive semidefinite $n$-by-$n$ matrix, $B$ is a real $m$-by-$n$ matrix with full rank $m \leq n$, and $B^T$ denotes the transposed matrix of $B$. Such systems typically result from the discretization of mixed variational problems for systems of partial differential equations (PDEs) (see Brezzi and Fortin [8]) in particular, from the discretization of optimization problems with PDE-constraints. A natural property of such a problem is that $A$ is positive definite on the kernel of $B$, i.e.,

$$(Aw, w) > 0 \quad \text{for all } w \in \ker B \text{ with } w \neq 0, \tag{1.2}$$

---

*Received by the editors May 26, 2006; accepted for publication (in revised form) by A. J. Wathen January 22, 2007; published electronically July 4, 2007.
http://www.siam.org/journals/simax/29-3/66097.html

[†] Center for Computational Engineering Science, RWTH Aachen, D-52074 Aachen, Germany. Current address: Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences, A-4040 Linz, Austria (schoeberl@mathcces.rwth-aachen.de, js@jku.at).

[‡] Institute of Computational Mathematics, Johannes Kepler University, A-4040 Linz, Austria (zulehner@numa.uni-linz.ac.at). The work of this author was supported in part by the Austrian Science Fund (FWF) under the grant SFB F013/F1309.

where $(x, w)$ denotes the Euclidean scalar product. This condition guarantees, in combination with the full rank of $B$, that the matrix

$$\mathcal{K} = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}$$

is nonsingular.

Under the assumptions stated above, the system (1.1) can be interpreted as the Karush–Kuhn–Tucker (KKT) conditions, which characterize the solution $x$ of the following constrained optimization problem (see, e.g., Fletcher [14]):

$$\text{Minimize} \quad J(x) \equiv \frac{1}{2}(Ax, x) - (f, x) \quad \text{subject to the constraints} \quad Bx = g$$

with associated Lagrangian parameter $p$.

Most of the work on efficient iterative methods for solving (1.1) has been done under the assumption that the matrix $A$ is positive definite not only on $\ker B$ but on the whole space $\mathbb{R}^n$, with the consequence that the (negative) Schur complement $S = BA^{-1}B^T$ is well defined. Most of the proposed methods can be viewed as preconditioned Richardson methods for (1.1) typically accelerated by a Krylov subspace method; see Saad and van der Vorst [23] for a review of iterative methods for linear systems. The discussed preconditioners for $\mathcal{K}$ are 2-by-2 block matrices $\hat{\mathcal{K}}$ depending on a preconditioner $\hat{A}$ for approximating $A$ and a preconditioner $\hat{S}$, which is either interpreted as an approximation of the Schur complement $S$ or as an approximation of the so-called inexact Schur complement $H = B\hat{A}^{-1}B^T$. Typical classes of such preconditioners which rely on a positive definite matrix $A$ are block diagonal preconditioners (see, e.g., Rusten and Winther [22], Silvester and Wathen [24]), block triangular preconditioners (originating from the classical Uzawa method [2]; see also, e.g., Elman and Golub [12], Bramble, Pasciak, and Vassilev [7]), symmetric indefinite preconditioners (see, e.g., Dyn and Ferguson [11], Bank, Welfert, and Yserentant [3], Rozložník and Simoncini [21], Al-Jeiroudi, Gondzio, and Hall [1], and Dollar [9]), and symmetric positive definite block (but not block diagonal) preconditioners; see Vassilevski and Lazarov [26]. Depending on the properties of the preconditioned systems, Krylov subspace methods either for symmetric indefinite or for nonsymmetric systems like MINRES, BiCG, or GMRES were proposed. In Bramble and Pasciak [6], a block triangular preconditioner was used in order to obtain a preconditioned system which is symmetric and positive definite and, therefore, can be solved by the conjugate gradient method (CG), which is usually considered the best or at least the best-understood Krylov subspace method. The block triangular preconditioner in [6] requires a symmetric and positive definite approximation $\hat{A}$ with $A - \hat{A}$ positive definite. In [21] an interesting equivalence between the right preconditioned simplified BiCG and a preconditioned conjugate gradient method (PCG) was obtained for the proposed indefinite preconditioner for a particular choice of the residuals. Yet another strategy to use CG was discussed, e.g., in Fischer et al. [13] and in Benzi and Simoncini [5], where the saddle point problem (1.1) was reformulated by multiplying the second block row by $-1$ leading to a positive stable but nonsymmetric system matrix.

In this paper, however, we will focus on systems where $A$ is positive definite in a stable way (to be specified later) only on $\ker B$, a typical situation for certain classes of optimization problems with PDE-constraints. One strategy is to enforce the definiteness on the whole space $\mathbb{R}^n$ by the augmented Lagrangian approach, where the matrix $A$ and the vector $f$ in (1.1) are replaced by a matrix of the form $A_W = A + B^T W B$ and a vector $f_W = f + B^T W g$, respectively, with an appropriate matrix $W$; see, e.g., Fortin and Glowinski [15]. This does not change the solution of the problem, and the new (1,1) block $A_W$ becomes positive definite if $W$ is properly chosen, e.g., if it is positive definite and all methods from above applied to the augmented system could be used, in principle. It is, however, a delicate issue to choose the matrix $W$ in order to obtain good convergence properties; see the discussions in Golub and Greif [16], Golub, Greif, and Varah [17]. Another approach is offered by a particular class of symmetric indefinite preconditioners; the so-called constraint preconditioners; see, e.g., Keller, Gould, and Wathen [20], Gould, Hribar, and Nocedal [18], and Dollar et al. [10]. These preconditioners are not restricted to the case of positive definite matrices $A$. For this class of preconditioners (projected), PCG was successfully used as an acceleration technique. One possible drawback of this class of preconditioners is the computational costs involved in the application of the preconditioner, where in some way or another some projection onto $\ker B$ has to be realized.

For a much more detailed discussion of available methods for saddle point problems, we refer to the review article by Benzi, Golub, and Liesen [4].

Here we will take a different approach and discuss preconditioners $\hat{\mathcal{K}}$ for the original system matrix $\mathcal{K}$ (without augmentation), which, nevertheless, also work well in the case that $A$ is positive definite only on the kernel of $B$. Under appropriate assumptions it will be shown that the preconditioned matrix $\hat{\mathcal{K}}^{-1}\mathcal{K}$ is even symmetric and positive definite in some appropriate scalar product. Therefore, CG acceleration can be applied. In contrast to Bramble and Pasciak [6], this new technique requires a symmetric and positive definite approximation $\hat{A}$ with $\hat{A} - A$ positive definite, which is easier to achieve and can also be applied if $A$ itself is only positive definite on the kernel of $B$.

An important field of applications are PDE-constrained optimization problems, in particular, optimal control problems; see, e.g., Tröltzsch [25]. It is typical for optimal control problems that the cost functional contains an extra regularization parameter. If discretized by an appropriate finite element method, the resulting KKT system is of the form (1.1), where the matrices $A$ and $B$ depend on the underlying subdivision, say with mesh size $h$, and on the regularization parameter, say $\nu$. For optimal control problems with elliptic state equations and distributed control, a special symmetric indefinite preconditioner will be constructed, and convergence rate estimates are given which are robust in $h$ as well as in $\nu$.

The paper is organized as follows: In section 2 the considered class of preconditioners is introduced and analyzed. Section 3 describes how the algebraic conditions for the preconditioners are linked to the conditions of Brezzi's theorem for mixed variational problems, and a general framework for constructing the preconditioners is sketched. In section 4 a problem from optimal control is discussed and preconditioners are constructed which are robust with respect to the mesh size as well as to the involved regularization parameter. Implementation issues are discussed in section 5 and numerical experiments are presented in section 6, followed by some concluding remarks.

Throughout the paper the following notations are used: $M < N$ ($N > M$) iff $N - M$ is positive definite, and $M \leq N$ ($N \geq M$) iff $N - M$ is positive semidefinite for symmetric matrices $M$ and $N$. For a symmetric and positive definite matrix $M$, the associated scalar product $(v, w)_M$ and norm $\|v\|_M$ are given by

$$(v, w)_M = (Mv, w) \quad \text{and} \quad \|v\|_M = (v, v)_M^{1/2},$$

where $(v, w)$ (without index) denotes the Euclidean scalar product. The Euclidean norm of a vector $v$ is denoted by $\|v\|$ (without index).

## 2. A class of symmetric and indefinite preconditioners.

A well-known class of preconditioners is given by

$$\hat{\mathcal{K}} = \begin{pmatrix} \hat{A} & B^T \\ B & B\hat{A}^{-1}B^T - \hat{S} \end{pmatrix},$$

where $\hat{A}$ and $\hat{S}$ are symmetric and positive definite matrices; see Bank, Welfert, and Yserentant [3]. More precisely, we will assume that $\hat{A}$ and $\hat{S}$ are preconditioners; i.e., efficient evaluations of $\hat{A}^{-1}s$ and $\hat{S}^{-1}t$ are available for given vectors $s$ and $t$.

We have the following factorization:

$$\hat{\mathcal{K}} = \begin{pmatrix} I & 0 \\ B\hat{A}^{-1} & I \end{pmatrix} \begin{pmatrix} \hat{A} & B^T \\ 0 & -\hat{S} \end{pmatrix},$$

which implies that $\hat{\mathcal{K}}$ is nonsingular and that the solution of a linear system

$$\hat{\mathcal{K}} \begin{pmatrix} w \\ q \end{pmatrix} = \begin{pmatrix} s \\ t \end{pmatrix}$$

reduces to the consecutive solution of the following three linear systems:

$$\hat{A}\hat{w} = s,$$

$$\hat{S}\,q = B\hat{w} - t,$$

$$\hat{A}w = s - B^T q.$$

So, one application of the preconditioner $\hat{\mathcal{K}}$ requires two applications of the preconditioner $\hat{A}$ and one application of the preconditioner $\hat{S}$.

In Bank, Welfert, and Yserentant [3] and later in Zulehner [27], this preconditioner has been analyzed for the case that $A$ is positive definite. One important part of the analysis easily carries over to the case considered here.

THEOREM 2.1. [illegible] $A \ge 0$ [illegible] (1.2) [illegible] $\operatorname{rank} B = m$ [illegible] $\hat{A} > 0$ [illegible] $\hat{S} > 0$

1. [illegible]

$$\hat{A} \ge A \qquad \hat{S} \le B\hat{A}^{-1}B^T, \tag{2.1}$$

[illegible] $\hat{\mathcal{K}}^{-1}\mathcal{K}$ [illegible]

2. [illegible]

$$\hat{A} > A \qquad \hat{S} < B\hat{A}^{-1}B^T, \tag{2.2}$$

[illegible] $\hat{\mathcal{K}}^{-1}\mathcal{K}$ [illegible]

$$\left( \begin{pmatrix} x \\ p \end{pmatrix}, \begin{pmatrix} w \\ q \end{pmatrix} \right)_{\mathcal{D}} = ((\hat{A} - A)x, w) + ((B\hat{A}^{-1}B^T - \hat{S})p, q). \tag{2.3}$$

[illegible]. Apply Theorem 5.2 from Zulehner [27] to the regularized matrices $A + \varepsilon I$ and $\hat{A} + \varepsilon I$ for $\varepsilon > 0$, take the limit $\varepsilon \to 0$, and observe that $\mathcal{K}$ is nonsingular. ☐

Estimates for the extreme eigenvalues of $\hat{\mathcal{K}}^{-1}\mathcal{K}$ were derived in Zulehner [27] under the assumption that $A$ is positive definite on the whole space. However, the estimate for the smallest eigenvalue degenerates, if directly applied to the case considered here. In this paper this gap will be closed.

First of all, we have to discuss reasonable assumptions on $\hat{A}$ and $\hat{S}$, which measure the quality of these preconditioners. Comparing the matrix $\mathcal{K}$ and the preconditioner $\hat{\mathcal{K}}$, it seems to be natural to consider $\hat{A}$ as an approximation to $A$ at least on $\ker B$ and to consider $\hat{S}$ as an approximation to the so-called inexact Schur complement $H$, given by

$$H = B\hat{A}^{-1}B^T.$$

Therefore, we assume that constants $\alpha > 0$ and $\beta > 0$ exist such that

$$(Aw, w) \geq \alpha\,(\hat{A}w, w) \quad \text{for all } w \in \ker B$$

and

$$B\hat{A}^{-1}B^T \leq \beta\,\hat{S}.$$

Observe that we will still require condition (2.1); therefore $\alpha \leq 1$ and $\beta \geq 1$. The closer $\alpha$ and $\beta$ are to 1 the better we expect the preconditioner $\hat{\mathcal{K}}$ will be. This results in the following theorem.

THEOREM 2.2. [illegible] $A \geq 0$ [illegible] (1.2) [illegible] $\operatorname{rank} B = m$ [illegible] $\hat{A} > 0$ [illegible] $\hat{S} > 0$ [illegible]

$$(Aw, w) \geq \alpha\,(\hat{A}w, w) \quad [illegible] \quad w \in \ker B \quad [illegible] \quad \hat{A} \geq A, \tag{2.4}$$

[illegible]

$$\hat{S} \leq B\hat{A}^{-1}B^T \leq \beta\,\hat{S} \tag{2.5}$$

[illegible] $\alpha$ [illegible] $\beta$ [illegible] $0 < \alpha \leq 1$ [illegible] $0 < \beta \leq 1$ [illegible]

$$\lambda_{max}(\hat{\mathcal{K}}^{-1}\mathcal{K}) \leq \beta + \sqrt{\beta^2 - \beta} = \beta\,(1 + \sqrt{1 - 1/\beta})$$

[illegible]

$$\begin{aligned}\lambda_{min}(\hat{\mathcal{K}}^{-1}\mathcal{K}) &\geq \frac{1}{2}\left[2 + \alpha - 1/\beta - \sqrt{(2 + \alpha - 1/\beta)^2 - 4\alpha}\right] \\ &\geq \alpha\left[\frac{2}{\sqrt{1 - 1/\beta} + \sqrt{5 - 1/\beta}}\right]^2 > 0.\end{aligned}$$

[illegible]. The upper bound directly follows from Theorem 5.2 in Zulehner [27] again by considering the regularized matrices $A + \varepsilon I$ and $\hat{A} + \varepsilon I$ for $\varepsilon > 0$ with $\varepsilon \to 0$.

For the lower bound we consider an eigenvalue $\lambda$ of the matrix $\hat{\mathcal{K}}^{-1}\mathcal{K}$:

$$\mathcal{K}\begin{pmatrix} x \\ p \end{pmatrix} = \lambda\,\hat{\mathcal{K}}\begin{pmatrix} x \\ p \end{pmatrix},$$

which is equivalent to the eigenvalue problem

$$\mathcal{K}\begin{pmatrix} x \\ p \end{pmatrix} = \mu\,\mathcal{D}\begin{pmatrix} x \\ p \end{pmatrix}$$

with

$$\lambda = \frac{\mu}{1+\mu} \quad \text{and} \quad \mathcal{D} = \hat{\mathcal{K}} - \mathcal{K} = \begin{pmatrix} \hat{A} - A & 0 \\ 0 & B\hat{A}^{-1}B^T - \hat{S} \end{pmatrix},$$

or, in an equivalent variational form,

$$\begin{aligned} (Ax, w) + (Bw, p) &= \mu\,((\hat{A} - A)x, w) && \text{for all } w \in \mathbb{R}^n, \\ (Bx, q) &= \mu\,((B\hat{A}^{-1}B^T - \hat{S})p, q) && \text{for all } q \in \mathbb{R}^m. \end{aligned}$$

Now, two cases are distinguished: First, for the case $\mu \le 0$, it follows that $\lambda = \mu/(1+\mu) > 1$, since $\lambda$ must be positive by Theorem 2.1. (The case $\mu = -1$ can be excluded, since $\hat{\mathcal{K}}$ is nonsingular.) So, in this case, the eigenvalues $\lambda$ are bounded from below by 1.

Next, we consider the remaining case $\mu > 0$. Let

$$W = \ker B, \quad W^\perp = \{x \in \mathbb{R}^n : (\hat{A}x, w) = 0 \text{ for all } w \in W\}.$$

Then there is a unique representation of $x$ of the following form:

$$x = x_1 + x_2 \quad \text{with } x_1 \in W \text{ and } x_2 \in W^\perp.$$

Now the variational form reads

$$\begin{aligned} (Ax_1, w_1) + (Ax_2, w_1) &= \mu\,\Big[((\hat{A} - A)x_1, w_1) - (Ax_2, w_1)\Big], \\ (Ax_1, w_2) + (Ax_2, w_2) + (Bw_2, p) &= \mu\,\Big[-(Ax_1, w_2) + ((\hat{A} - A)x_2, w_2)\Big], \\ (Bx_2, q) &= \mu\,((B\hat{A}^{-1}B^T - \hat{S})p, q) \end{aligned}$$

for all $w_1 \in W$, $w_2 \in W^\perp$, $q \in \mathbb{R}^m$. From the first equation we obtain for $w_1 = x_1$ that

$$\alpha\,(x_1, x_1)_{\hat{A}} \le (Ax_1, x_1) = \mu\,((\hat{A} - A)x_1, x_1) - (\mu + 1)(Ax_2, x_1).$$

Using

$$\begin{aligned} |(Aw_2, w_1)| = |((\hat{A} - A)w_2, w_1)| &\le ((\hat{A} - A)w_1, w_1)^{1/2}((\hat{A} - A)w_2, w_2)^{1/2} \\ &\le \sqrt{1-\alpha}\,\|w_1\|_{\hat{A}}\,\|w_2\|_{\hat{A}} \quad \text{for all } w_1 \in W,\ w_2 \in W^\perp, \end{aligned}$$

it follows that

$$\alpha\,(x_1, x_1)_{\hat{A}} \le \mu\,(1-\alpha)\,(x_1, x_1)_{\hat{A}} + (\mu + 1)\sqrt{1-\alpha}\,\|x_1\|_{\hat{A}}\|x_2\|_{\hat{A}},$$

which implies

$$\alpha\,\|x_1\|_{\hat{A}} \le \mu\,(1-\alpha)\,\|x_1\|_{\hat{A}} + (\mu + 1)\sqrt{1-\alpha}\,\|x_2\|_{\hat{A}}.$$

From the second equation we obtain

$$\sup_{w_2 \in W^\perp} \frac{(Bw_2, p)}{\|w_2\|_{\hat{A}}} = \sup_{w_2 \in W^\perp} \frac{-(\mu+1)(Ax_1, w_2) + ((\mu(\hat{A}-A)-A)x_2, w_2)}{\|w_2\|_{\hat{A}}}.$$

Using

$$|(Ax_1, w_2)| = |(Aw_2, x_1)| \le \sqrt{1-\alpha}\, \|x_1\|_{\hat{A}}\, \|w_2\|_{\hat{A}}$$

and

$$\begin{aligned} |([\mu(\hat{A}-A)-A]x_2, w_2)| &= |(\hat{A}^{-1}[\mu(\hat{A}-A)-A]x_2, w_2)_{\hat{A}}| \\ &\le \|\hat{A}^{-1}[\mu(\hat{A}-A)-A]\|_{\hat{A}} \|x_2\|_{\hat{A}}\, \|w_2\|_{\hat{A}} \end{aligned}$$

with

$$\|\hat{A}^{-1}[\mu(\hat{A}-A)-A]\|_{\hat{A}} \le \mu\, \|\hat{A}^{-1}(\hat{A}-A)\|_{\hat{A}} + \|\hat{A}^{-1}A\|_{\hat{A}} \le \mu+1,$$

it follows that

$$\sup_{w_2 \in W^\perp} \frac{(Bw_2, p)}{\|w_2\|_{\hat{A}}} \le (\mu+1)\sqrt{1-\alpha}\, \|x_1\|_{\hat{A}} + (\mu+1)\, \|x_2\|_{\hat{A}}.$$

From the third equation we obtain

$$\sup_{0 \ne q} \frac{(Bx_2, q)}{\|q\|_H} = \sup_{0 \ne q} \frac{\mu\, ((B\hat{A}^{-1}B^T - \hat{S})p, q)}{\|q\|_H} \le \mu\, (1 - 1/\beta)\, \|p\|_H.$$

Observe that, for the left-hand sides of the last two inequalities, we have the following well-known representations:

$$\sup_{w_2 \in W^\perp} \frac{(Bw_2, p)}{\|w_2\|_{\hat{A}}} = \sup_{w \in \mathbb{R}^n} \frac{(Bw, p)}{\|w\|_{\hat{A}}} = (B\hat{A}^{-1}B^T p, p)^{1/2} = \|p\|_H$$

and

$$\begin{aligned} \sup_{0 \ne q \in \mathbb{R}^m} \frac{(Bx_2, q)}{\|q\|_H} &= (B^T H^{-1} B x_2, x_2)^{1/2} = (\hat{A}^{-1} B^T H^{-1} B x_2, x_2)_{\hat{A}}^{1/2} \\ &= (x_2, x_2)_{\hat{A}}^{1/2} = \|x_2\|_{\hat{A}}, \end{aligned}$$

since $P = \hat{A}^{-1}B^T H^{-1} B$ is a projection onto $W^\perp$, so $Px_2 = x_2$ for $x_2 \in W^\perp$.

Hence, in summary,

$$\underbrace{\begin{pmatrix} \alpha & -\sqrt{1-\alpha} & 0 \\ -\sqrt{1-\alpha} & -1 & 1 \\ 0 & 1 & 0 \end{pmatrix}}_{K} \underbrace{\begin{pmatrix} \|x_1\|_{\hat{A}} \\ \|x_2\|_{\hat{A}} \\ \|p\|_H \end{pmatrix}}_{e}$$

$$\le \mu \underbrace{\begin{pmatrix} 1-\alpha & \sqrt{1-\alpha} & 0 \\ \sqrt{1-\alpha} & 1 & 0 \\ 0 & 0 & 1-1/\beta \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} \|x_1\|_{\hat{A}} \\ \|x_2\|_{\hat{A}} \\ \|p\|_H \end{pmatrix}}_{e}. \tag{2.6}$$

Since $K^{-1}$ is nonnegative elementwise, it follows that

$$e \le \mu\, K^{-1} D e.$$

Elementary calculations show that

$$\nu_+ = \frac{1}{2\alpha}\left[2 - \alpha - 1/\beta + \sqrt{(2 - \alpha - 1/\beta)^2 + 4\alpha(1 - 1/\beta)}\right]$$

is a nonnegative eigenvalue of $K^{-1}D$ with componentwise nonnegative left eigenvector $l_+^T$, given by

$$l_+^T = \left(\sqrt{1-\alpha}, 1, \alpha\nu_+ - 1 + \alpha\right).$$

Then

$$l_+^T e \le \mu\nu_+ l_+^T e.$$

Obviously, $l_+^T e \ge 0$. One can easily show that $\nu_+ > 0$ and $l_+^T e > 0$: $\nu_+ = 0$ implies $\alpha = \beta = 1$, then (2.6) implies $e = 0$. In a similar way the case $l_+^T e = 0$ can be excluded.

Therefore, after dividing by $l_+^T e > 0$, we obtain

$$\mu \ge \frac{1}{\nu_+}.$$

Consequently,

$$\begin{aligned}\lambda &= \frac{\mu}{1+\mu} \ge \frac{1}{1+\nu_+} = \frac{1}{2}\left[2 + \alpha - 1/\beta - \sqrt{(2 + \alpha - 1/\beta)^2 - 4\alpha}\right] \\ &= \frac{2\alpha}{2 + \alpha - 1/\beta + \sqrt{(2 + \alpha - 1/\beta)^2 - 4\alpha}} \\ &\ge \frac{2\alpha}{3 - 1/\beta + \sqrt{(3 - 1/\beta)^2 - 4}} = \alpha\left[\frac{2}{\sqrt{1 - 1/\beta} + \sqrt{5 - 1/\beta}}\right]^2 > 0.\end{aligned}$$

This lower bound is obviously smaller than 1, which was the lower bound for the first case $\mu \le 0$. This completes the proof. ∎

By slightly strengthening the conditions (2.4) and (2.5) to

$$(Aw, w) \ge \alpha\,(\hat{A}w, w) \quad \text{for all } w \in \ker B \quad \text{and} \quad \hat{A} > A \tag{2.7}$$

and

$$\hat{S} < B\hat{A}^{-1}B^T \le \beta\,\hat{S}, \tag{2.8}$$

the scalar product (2.3) is well defined, and, by Theorem 2.1, the standard CG can be applied to the preconditioned system

$$\hat{\mathcal{K}}^{-1}\mathcal{K}\begin{pmatrix} x \\ p \end{pmatrix} = \hat{\mathcal{K}}^{-1}\begin{pmatrix} f \\ g \end{pmatrix} \tag{2.9}$$

with respect to the scalar product (2.3).

The actual construction of the preconditioners $\hat{A}$ and $\hat{S}$ is usually done in two steps. First, some preliminary candidates $\hat{A}_0$ and $\hat{S}_0$ are chosen which approximate the matrices $A$ and $B\hat{A}_0^{-1}B^T$. In the second step, these candidates are properly scaled: $\hat{A} = (1/\sigma)\,\hat{A}_0$ and $\hat{S} = (\sigma/\tau)\,\hat{S}_0$, where the positive parameters $\sigma$ and $\tau$ must be chosen such that (2.2) are satisfied, i.e.,

$$\frac{1}{\sigma}\,\hat{A}_0 > A \quad \text{and} \quad \frac{1}{\tau}\,\hat{S}_0 < B\hat{A}_0^{-1}B^T.$$

So, the correct choice of the parameters $\sigma$ and $\tau$ requires some rough information of the size of the largest eigenvalue of $A$ relative to $\hat{A}_0$, which is, in general, quite easy to obtain and of the size of the smallest eigenvalue of $B\hat{A}_0^{-1}B^T$ relative to $\hat{S}_0$, which, in general, is more costly, but which is available here from the analysis for the problem discussed in section 4. The values of $\alpha$ and $\beta$ in (2.7) and (2.8) are not needed for the construction, but only for the analysis.

It is well known (see, e.g., Hackbusch [19]) that the error $e^{(k)}$ for the $k$th iterate $(x^{(k)}, p^{(k)})^T$ measured in the corresponding energy norm can be estimated by

$$e^{(k)} \le \frac{2q^k}{1+q^{2k}}\,e^{(0)} \quad \text{with} \quad q = \frac{\sqrt{\kappa(\hat{\mathcal{K}}^{-1}\mathcal{K})}-1}{\sqrt{\kappa(\hat{\mathcal{K}}^{-1}\mathcal{K})}+1},$$

where $\kappa(\hat{\mathcal{K}}^{-1}\mathcal{K})$ denotes the relative condition number

$$\kappa(\hat{\mathcal{K}}^{-1}\mathcal{K}) = \frac{\lambda_{\max}(\hat{\mathcal{K}}^{-1}\mathcal{K})}{\lambda_{\min}(\hat{\mathcal{K}}^{-1}\mathcal{K})}.$$

From Theorem 2.2 the following upper bound for the relative condition number follows:

$$\begin{aligned}\kappa(\hat{\mathcal{K}}^{-1}\mathcal{K}) &\le \frac{2(\beta+\sqrt{\beta^2-\beta})}{2+\alpha-1/\beta-\sqrt{(2+\alpha-1/\beta)^2-4\alpha}} \equiv \kappa(\alpha,\beta)\\ &\le \frac{\beta}{\alpha}\,(1+\sqrt{1-1/\beta})\left[\frac{\sqrt{1-1/\beta}+\sqrt{5-1/\beta}}{2}\right]^2.\end{aligned}$$

This shows that the convergence rate $q$ can be bounded by $\alpha$ and $\beta$ only. If the preconditioners are chosen such that $\alpha$ and $\beta$ are independent of certain parameters like the mesh size $h$ of some discretization or some involved regularization parameter $\nu$, then the convergence rate is also robust with respect to such parameters.

Furthermore, for $\alpha \to 1$ and $\beta \to 1$, the lower and upper bounds for the eigenvalues in Theorem 2.2 both approach 1 (implying that all eigenvalues of the preconditioned matrix $\hat{\mathcal{K}}^{-1}\mathcal{K}$ approach 1), leading to a relative condition number approaching 1 and a convergence factor $q$ approaching 0.

In the limit case $\alpha = 1$ and $\beta = 1$, one can easily derive the following representations for the preconditioners from the conditions (2.4) and (2.5):

$$\hat{A} = A + B^T\,W\,B \quad \text{and} \quad \hat{S} = B\hat{A}^{-1}B^T$$

for some matrix $W \ge 0$. Then, we obtain:

$$\hat{\mathcal{K}} = \begin{pmatrix} A + B^TWB & B^T \\ B & 0 \end{pmatrix}.$$

From the previous considerations, it follows in this case that all eigenvalues of $\hat{\mathcal{K}}^{-1}\mathcal{K}$ must be equal to 1. Moreover, it can easily be shown that

$$\left[I - \hat{\mathcal{K}}^{-1}\mathcal{K}\right]^2 = 0.$$

So, the corresponding preconditioned Richardson method terminates at the solution after two steps.

In a simplified way one could describe the proposed strategy as follows: Good preconditioners $\hat{A}$ can be interpreted as good approximations to some augmented matrix $A + B^T W B$, but we do not change the matrix $A$ itself in the system matrix $\mathcal{K}$. This seems to be only a slight variant to the augmented Lagrangian approach, where first $A$ itself is replaced by $A + B^T W B$ in $\mathcal{K}$. However, the actual construction of the preconditioner is not based on first selecting some augmentation matrix $W$ and then preconditioning the augmented matrix. Instead, as will be detailed in the next section, the construction is guided by the analysis of an underlying (infinite-dimensional) variational problem, whose discretization leads to the discussed large scale linear systems of equations in saddle point form.

## 3. Application to mixed variational problems.

Consider an (infinite-dimensional) mixed variational problem of the following form: Find $x \in X$ and $p \in Q$ such that

$$\begin{aligned} a(x, w) \ + \ b(w, p) &= \langle F, w\rangle \quad \text{for all } w \in X, \\ b(x, q) &= \langle G, q\rangle \quad \text{for all } q \in Q. \end{aligned}$$

Here, $X$ and $Q$ are real Hilbert spaces, $a : X \times X \longrightarrow \mathbb{R}$ and $b : X \times Q \longrightarrow \mathbb{R}$ are bilinear forms, $F : X \longrightarrow \mathbb{R}$ and $G : Q \longrightarrow \mathbb{R}$ are continuous linear functionals, and $\langle F, w\rangle$ ($\langle G, q\rangle$) denotes the evaluation of $F$ ($G$) at the element $w$ ($q$).

The existence and uniqueness of a solution to this mixed variational problem is well established (Brezzi's theorem; see Brezzi and Fortin [8]) under the following conditions:

1. The bilinear form $a$ is bounded:

$$a(x, w) \le \|a\| \, \|x\|_X \|w\|_X \quad \text{for all } x, w \in X.$$

2. The bilinear form $a$ is coercive on $\ker B = \{w \in X : b(w, q) = 0 \text{ for all } q \in Q\}$: There exists a constant $\alpha_0 > 0$ such that

$$a(w, w) \ge \alpha_0 \, \|w\|_X^2 \quad \text{for all } w \in \ker B.$$

3. The bilinear form $b$ is bounded:

$$\sup_{0 \ne w \in X} \frac{b(w, q)}{\|w\|_X} \le \|b\| \, \|q\|_Q \quad \text{for all } q \in Q.$$

4. The bilinear form $b$ satisfies the inf-sup condition: There exists a constant $k_0 > 0$ such that

$$\sup_{0 \ne w \in X} \frac{b(w, q)}{\|w\|_X} \ge k_0 \, \|q\|_Q \quad \text{for all } q \in Q.$$

Under the additional assumptions that

5. the bilinear form $a$ is symmetric on $X$:

$$a(x, w) = a(w, x) \quad \text{for all } x, w \in X, \text{ and}$$

6. the bilinear form $a$ is nonnegative on $X$:

$$a(w, w) \geq 0 \quad \text{for all } w \in X,$$

Brezzi's theorem implies the equivalence of the mixed variational problem to the following constrained optimization problem: Find $x \in X$ such that

$$J(x) = \min_{w \in X_g} J(w) \tag{3.1}$$

with

$$J(w) = \frac{1}{2} a(w, w) - \langle F, w \rangle$$

and

$$X_g = \{w \in X : b(w, q) = \langle G, q \rangle \text{ for all } q \in Q\}.$$

For discretizing the infinite-dimensional problem the spaces $X$ and $Q$ are replaced by finite-dimensional subspaces $X_h \subset X$ and $Q_h \subset Q$, which results in the following finite-dimensional variational problem: Find $x_h \in X_h$ and $p_h \in Q_h$ such that

$$\begin{aligned} a(x_h, w_h) \;+\; b(w_h, p_h) &= \langle F, w_h \rangle \quad && \text{for all } w_h \in X_h, \\ b(x_h, q_h) &= 0 && \text{for all } q_h \in Q_h. \end{aligned}$$

By introducing suitable basis functions in $X_h$ and $Q_h$, we finally obtain the following saddle point problem in matrix-vector notation:

$$\begin{aligned} A_h \underline{x}_h \;+\; B_h^T \underline{p}_h &= \underline{f}_h, \\ B_h \underline{x}_h &= \underline{g}_h, \end{aligned}$$

where $\underline{x}_h$ and $\underline{p}_h$ denote the corresponding vectors of coefficients with respect to these basis functions.

We assume that the conditions of Brezzi's theorem are also satisfied in $X_h$ and $Q_h$. This is trivial for the first and third conditions. The second and fourth conditions must be proven for the particular equations and elements. To simplify the notation the same symbols are used to denote the constants.

The scalar products $(x, q)_X$ and $(p, q)_Q$ are bilinear forms on $X_h$ and $Q_h$. The associated matrices representing these scalar products are denoted by $\underline{X}_h$ and $\underline{Q}_h$, respectively, i.e.,

$$(x_h, w_h)_X = (\underline{X}_h \underline{x}_h, \underline{w}_h), \quad (p_h, q_h)_Q = (\underline{Q}_h \underline{p}_h, \underline{q}_h).$$

Using matrix-vector notations, the conditions of Brezzi's theorem on $X_h$ and $Q_h$ are

$$A_h \leq \|a\| \, \underline{X}_h, \tag{3.2}$$

$$(A_h, \underline{w}_h, \underline{w}_h) \geq \alpha_0 \, (\underline{X}_h \underline{w}_h, \underline{w}_h) \quad \text{for all } \underline{w}_h \in \ker B_h, \tag{3.3}$$

$$B_h \underline{X}_h^{-1} B_h^T \leq \|b\|^2 \, \underline{Q}_h, \tag{3.4}$$

$$B_h \underline{X}_h^{-1} B_h^T \geq k_0^2 \, \underline{Q}_h. \tag{3.5}$$

For the third and fourth condition we used the well-known representation

$$\sup_{0 \neq w_h \in X_h} \frac{b(w_h, q_h)}{\|w_h\|_X} = (B_h \underline{X}_h^{-1} B_h^T \underline{q}_h, \underline{q}_h)^{1/2}.$$

Comparing (3.2)–(3.5) with the conditions (2.7) and (2.8) it seems to be reasonable to choose for $\hat{A}_h$ a suitable multiple of the matrix $\underline{X}_h$ and for $\hat{S}_h$ a suitable multiple of the matrix $\underline{Q}_h$. However, since the application of the preconditioner $\hat{\mathcal{K}}_h$ requires the solution of linear systems with the matrices $\hat{A}_h$ and $\hat{S}_h$, this would require the inversion of these matrices $\underline{X}_h$ and $\underline{Q}_h$. In typical applications (see the next section) (parts of) $\underline{X}_h$ and $\underline{Q}_h$ are the stiffness matrices of second order differential operators. So, the exact inversion could be too costly. Therefore, it is recommended to use approximations, say $\hat{X}_h$ and $\hat{Q}_h$, that are easy to invert (i.e., preconditioners) instead of $\underline{X}_h$ and $\underline{Q}_h$:

$$\hat{A}_h = \frac{1}{\sigma} \hat{X}_h \quad \text{and} \quad \hat{S}_h = \frac{\sigma}{\tau} \hat{Q}_h \tag{3.6}$$

for some real parameters $\sigma > 0$ and $\tau > 0$, which are needed for a suitable scaling. We assume that the quality of these preconditioners can be described by spectral estimates, e.g., of the form

$$(1 - q_X) \hat{X}_h \leq \underline{X}_h \leq \hat{X}_h \quad \text{and} \quad (1 - q_Q) \hat{Q}_h \leq \underline{Q}_h \leq \hat{Q}_h \tag{3.7}$$

with constants $q_X, q_Q \in [0, 1)$. The smaller these constants are the better the preconditioners $\hat{X}_h$ and $\hat{Q}_h$ approximate the matrices $\underline{X}_h$ and $\underline{Q}_h$.

Combining all estimates we easily obtain the following lemma.

LEMMA 3.1. [illegible] (3.2)–(3.7) [illegible] (2.7) [illegible] (2.8) [illegible]

$$\alpha = \sigma (1 - q_X) \alpha_0 \quad [illegible] \quad \beta = \tau \|b\|^2$$

[illegible] $\sigma$ [illegible] $\tau$ [illegible]

$$\sigma < \frac{1}{\|a\|} \quad [illegible] \quad \tau > \frac{1}{(1 - q_X)(1 - q_Q) k_0^2}.$$

[illegible]. We have

$$A_h \leq \|a\| \underline{X}_h \leq \|a\| \hat{X}_h = \sigma \|a\| \hat{A}_h < \hat{A}_h$$

if $\sigma < 1/\|a\|$. Next

$$(A_h \underline{w}_h, \underline{w}_h) \geq \alpha_0 (\underline{X}_h \underline{w}_h, \underline{w}_h) \geq (1 - q_X) \alpha_0 (\hat{X}_h \underline{w}_h, \underline{w}_h) = \alpha (\hat{A}_h \underline{w}_h, \underline{w}_h)$$

with $\alpha = \sigma (1 - q_X) \alpha_0$. Next

$$B_h \hat{A}_h^{-1} B_h^T = \sigma B_h \hat{X}_h^{-1} B_h^T \leq \sigma B_h \underline{X}_h^{-1} B_h^T \leq \sigma \|b\|^2 \underline{Q}_h \leq \sigma \|b\|^2 \hat{Q}_h = \beta \hat{S}_h$$

with $\beta = \tau \|b\|^2$. Finally

$$\begin{aligned} B_h \hat{A}_h^{-1} B_h^T &= \sigma B_h \hat{X}_h^{-1} B_h^T \geq \sigma (1 - q_X) B_h \underline{X}_h^{-1} B_h^T \geq \sigma (1 - q_X) k_0^2 \underline{Q}_h \\ &\geq \sigma (1 - q_X) (1 - q_Q) k_0^2 \hat{Q}_h = \tau (1 - q_X) (1 - q_Q) k_0^2 \hat{S}_h > \hat{S}_h \end{aligned}$$

if $\tau > 1/[(1-q_X)(1-q_Q)k_0^2]$. ☐

Good and efficient preconditioners $\hat{X}_h$ and $\hat{Q}_h$ are usually available, as will be shown for a particular problem in the next section. Therefore, the quantities $q_X$ and $q_Q$ are typically small, say 0.1.

Roughly speaking, the parameter $\sigma$ has to be sufficiently small, while the parameter $\tau$ has to be sufficiently large in order to guarantee the conditions (2.7) and (2.8). On the other hand, in order to obtain a small upper bound $\kappa(\alpha, \beta)$ for the condition number of the preconditioned matrix $\hat{\mathcal{K}}^{-1}\mathcal{K}$, $\alpha$ should be as large as possible and $\beta$ should be as small as possible, i.e., $\sigma$ should be as large as possible and $\tau$ should be as small as possible. This, of course, requires at least a rough quantitative knowledge of the constants $\|a\|$ and $k_0$, which are involved in the choice of $\sigma$ and $\tau$.

Next, we will study a particular problem from optimal control, where the parameters $\|a\|$, $\alpha_0$, $\|b\|$, and $k_0$ are known.

## 4. A problem from optimal control.

Let $\Omega \subset \mathbb{R}^d$ be an open and bounded set. We consider the following optimization problem with PDE-constraints: Find the state $y \in H^1(\Omega)$ and the control $u \in L^2(\Omega)$ such that

$$J(y,u) = \min_{(z,v)\in H^1(\Omega)\times L^2(\Omega)} J(z,v)$$

subject to the state equation with distributed control $u$

$$\begin{aligned} -\Delta y + y &= u \quad \text{in } \Omega, \\ \frac{\partial y}{\partial n} &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

where the cost functional is given by

$$J(y,u) = \frac{1}{2}\int_\Omega (y - y_d)^2 \, dx + \frac{\nu}{2}\int_\Omega u^2 \, dx.$$

More precisely, we prescribe the state equation in weak form:

$$\int_\Omega \nabla y \cdot \nabla q \, dx + \int_\Omega y\, q \, dx = \int_\Omega u\, q \, dx \quad \text{for all } q \in H^1(\Omega).$$

Let $X = Y \times U$ with $Y = H^1(\Omega)$, $U = L^2(\Omega)$, and $Q = H^1(\Omega)$. With $x = (y,u) \in X$, $w = (z,v) \in X$, and $q \in Q$, we introduce the following bilinear forms and linear functionals:

$$\begin{aligned} a(x,w) &= \int_\Omega y\, z \, dx + \nu \int_\Omega u\, v \, dx, \\ b(w,q) &= \int_\Omega \nabla z \cdot \nabla q \, dx + \int_\Omega z\, q \, dx - \int_\Omega v\, q \, dx, \\ \langle F, w\rangle &= \int_\Omega y_d\, z \, dx, \\ \langle G, q\rangle &= 0. \end{aligned}$$

With this setting the optimization problem is of the standard form (3.1).

The conditions of Brezzi's theorem can easily be verified for the Hilbert spaces $X = Y \times U$ and $Q$ introduced above and equipped with the standard scalar products $(y,z)_{H^1(\Omega)}$ in $Y$, $(u,v)_{L^2(\Omega)}$ in $U$, and $(p,q)_{H^1(\Omega)}$ in $Q$. Then, however, the parameters $\|a\|$, $\alpha_0$, $\|b\|$, and $k_0$ depend on the regularization parameter $\nu$, eventually resulting in convergence rates also depending on $\nu$.

With a different scaling of the scalar products in $Y$, $U$, and $Q$ we obtain parameters $\|a\|$, $\alpha_0$, $\|b\|$, and $k_0$ independent of $\nu$, eventually leading to preconditioners with convergence rates robust in $\nu$. In particular, we consider the following new scalar products $(y,z)_Y$ in $Y = H^1(\Omega)$, $(u,v)_U$ in $U = L^2(\Omega)$, and $(p,q)_Q$ in $Q = H^1(\Omega)$:

$$(y,z)_Y = (y,z)_{L^2(\Omega)} + \sqrt{\nu}\,(y,z)_{H^1(\Omega)}, \quad (u,v)_U = \nu\,(u,v)_{L^2(\Omega)},$$

and

$$(p,q)_Q = \frac{1}{\nu}\,(p,q)_{L^2(\Omega)} + \frac{1}{\sqrt{\nu}}\,(p,q)_{H^1(\Omega)},$$

and we set $(x,w)_X = (y,z)_Y + (u,v)_U$ for $x = (y,u), w = (z,v) \in X = Y \times U$. Observe that the corresponding new norms are equivalent to the standard norms in these spaces for fixed $\nu > 0$.

With these definitions of the scalar products the following properties can be verified.

LEMMA 4.1.

1. [illegible] $a$ [illegible]

$$a(x,w) \le \|x\|_X \|w\|_X \quad \text{[illegible]} \quad x, w \in X.$$

2. [illegible] $a$ [illegible] $\ker B$

$$a(w,w) \ge \alpha_0\,\|w\|_X^2 \quad \text{[illegible]} \quad w \in \ker B \quad \text{[illegible]} \quad \alpha_0 = \frac{2}{3}.$$

3. [illegible] $b$ [illegible]

$$\sup_{0 \ne w \in X} \frac{b(w,q)}{\|w\|_X} \le \|q\|_Q \quad \text{[illegible]} \quad q \in Q.$$

4. [illegible] $b$ [illegible] inf-sup [illegible]

$$\sup_{0 \ne w \in X} \frac{b(w,q)}{\|w\|_X} \ge k_0\,\|q\|_Q \quad \text{[illegible]} \quad k_0 = \sqrt{\frac{3}{4}}.$$

[illegible] 1 of Lemma 4.1 is trivial since $a$ is symmetric and $a(w,w) \le \|w\|_X^2$. For 2 take $w = (z,v) \in \ker B$. Then

$$(z,q)_{H^1(\Omega)} = (v,q)_{L^2(\Omega)} \quad \text{for all } q \in H^1(\Omega).$$

In particular, it follows for $q = z$ that

$$\|z\|_{H^1(\Omega)}^2 = (v,z)_{L^2(\Omega)} \le \|v\|_{L^2(\Omega)} \|z\|_{L^2(\Omega)},$$

which implies

$$\|w\|_X^2 = \|z\|_Y^2 + \|v\|_U^2 \le \|z\|_{L^2(\Omega)}^2 + \sqrt{\nu}\,\|z\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \nu\,\|v\|_{L^2(\Omega)}^2.$$

Then

$$a(w,w) \ge \alpha_0 \, \|w\|_X^2$$

is certainly satisfied if

$$a(w,w) \ge \alpha_0 \, \left[ \|z\|_{L^2(\Omega)}^2 + \sqrt{\nu} \, \|z\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \nu \, \|v\|_{L^2(\Omega)}^2 \right],$$

which is equivalent to

$$(1-\alpha_0) \, \|z\|_{L^2(\Omega)}^2 - \alpha_0 \sqrt{\nu} \, \|z\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + (1-\alpha_0) \, \nu \, \|v\|_{L^2(\Omega)}^2 \ge 0.$$

This is obviously the case for $\alpha_0 = 2/3$, since

$$\frac{1}{3} \|z\|_{L^2(\Omega)}^2 - \frac{2}{3} \sqrt{\nu} \, \|z\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \frac{1}{3} \nu \, \|v\|_{L^2(\Omega)}^2 = \frac{1}{3} \left[ \|z\|_{L^2(\Omega)} - \sqrt{\nu} \|v\|_{L^2(\Omega)} \right]^2.$$

To show 3 and 4, we start with the following formula:

$$\begin{aligned}
\sup_{0 \ne w \in X} \frac{b(w,q)^2}{\|w\|_X^2} &= \sup_{0 \ne (z,v) \in Y \times U} \frac{\left[ (z,q)_{H^1(\Omega)} - (v,q)_{L^2(\Omega)} \right]^2}{\|z\|_Y^2 + \|v\|_U^2} \\
&= \sup_{0 \ne z \in Y} \frac{(z,q)_{H^1(\Omega)}^2}{\|z\|_Y^2} + \sup_{0 \ne v \in U} \frac{(v,q)_{L^2(\Omega)}^2}{\|v\|_U^2} \\
&= \sup_{0 \ne z \in Y} \frac{(z,q)_{H^1(\Omega)}^2}{\|z\|_Y^2} + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2.
\end{aligned}$$

Then 3 easily follows from the estimates

$$\begin{aligned}
\sup_{0 \ne z \in Y} \frac{(z,q)_{H^1(\Omega)}^2}{\|z\|_Y^2} + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2 &\le \sup_{0 \ne z \in Y} \frac{\|z\|_{H^1(\Omega)}^2 \|q\|_{H^1(\Omega)}^2}{\|z\|_Y^2} + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2 \\
&= \sup_{0 \ne z \in Y} \frac{\|z\|_{H^1(\Omega)}^2 \|q\|_{H^1(\Omega)}^2}{\|z\|_{L^2(\Omega)}^2 + \sqrt{\nu} \, \|z\|_{H^1(\Omega)}^2} + \frac{1}{\nu} \|q\|_{L^2(\Omega)}^2 \\
&\le \frac{1}{\sqrt{\nu}} \|q\|_{H^1(\Omega)}^2 + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2 = \|q\|_Q^2.
\end{aligned}$$

For 4 observe that

$$\begin{aligned}
\sup_{0 \ne z \in Y} \frac{(z,q)_{H^1(\Omega)}^2}{\|z\|_Y^2} + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2 &\ge \frac{\|q\|_{H^1(\Omega)}^4}{\|q\|_Y^2} + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2 \\
&= \frac{\|q\|_{H^1(\Omega)}^4}{\|q\|_{L^2(\Omega)}^2 + \sqrt{\nu} \, \|q\|_{H^1(\Omega)}^2} + \frac{1}{\nu} \, \|q\|_{L^2(\Omega)}^2.
\end{aligned}$$

Then the inf-sup condition

$$\sup_{0 \ne w \in X} \frac{b(w,q)}{\|w\|_X} \ge k_0 \, \|q\|_Q$$

is certainly satisfied if

$$\frac{\|q\|^4_{H^1(\Omega)}}{\|q\|^2_{L^2(\Omega)} + \sqrt{\nu}\,\|q\|^2_{H^1(\Omega)}} + \frac{1}{\nu}\,\|q\|^2_{L^2(\Omega)} \geq k_0^2\,\|q\|^2_Q = k_0^2\left[\frac{1}{\nu}\,\|q\|^2_{L^2(\Omega)} + \frac{1}{\sqrt{\nu}}\,\|q\|^2_{H^1(\Omega)}\right],$$

which is equivalent to

$$(1-k_0^2)\,\|q\|^4_{H^1(\Omega)} + (1-2k_0^2)\frac{1}{\sqrt{\nu}}\,\|q\|^2_{L^2(\Omega)}\|q\|^2_{H^1(\Omega)} + (1-k_0^2)\frac{1}{\nu}\,\|q\|^4_{L^2(\Omega)} \geq 0.$$

This is obviously the case for $k_0^2 = 3/4$ since

$$\begin{aligned}&\frac{1}{4}\,\|q\|^4_{H^1(\Omega)} - \frac{1}{2}\frac{1}{\sqrt{\nu}}\,\|q\|^2_{L^2(\Omega)}\|q\|^2_{H^1(\Omega)} + \frac{1}{4}\frac{1}{\nu}\,\|q\|^4_{L^2(\Omega)}\\ &\quad = \frac{1}{4}\left[\|q\|^2_{H^1(\Omega)} - \frac{1}{\sqrt{\nu}}\,\|q\|^2_{L^2(\Omega)}\right]^2. \qquad \square\end{aligned}$$

By Brezzi's theorem it now follows that the optimization problem is equivalent to the following mixed variational problem: Find $x \in H^1(\Omega) \times L^2(\Omega)$ and $p \in H^1(\Omega)$ such that

$$\begin{aligned} a(x,w) \;+\; b(w,p) &= \langle F, x\rangle \quad \text{for all } w \in H^1(\Omega)\times L^2(\Omega),\\ b(x,q) &= 0 \qquad\quad\; \text{for all } q \in H^1(\Omega).\end{aligned}$$

For the spaces $Y_h = U_h = Q_h$ we choose, as an example, the space of piecewise linear and continuous functions on a simplicial subdivision of $\Omega$. By introducing the standard nodal basis, we finally obtain the following saddle point problem in matrix-vector notation:

$$\begin{aligned} A_h\underline{x}_h \;+\; B_h^T\underline{p}_h &= \underline{f}_h,\\ B_h\underline{x}_h &= 0,\end{aligned}$$

with

$$A_h = \begin{pmatrix} M_h & 0\\ 0 & \nu\,M_h\end{pmatrix} \quad\text{and}\quad B_h = \begin{pmatrix} K_h & -M_h\end{pmatrix},$$

where $M_h$ denotes the mass matrix representing the $L^2(\Omega)$ inner product on $Y_h$ and $K_h$ denotes the stiffness matrix representing the bilinear form (on $Y$) of the state equation, here $(\nabla y, \nabla q)_{L^2(\Omega)} + (y,q)_{L^2(\Omega)}$, on $Y_h$.

For the matrices $\underline{X}_h$ and $\underline{Q}_h$ representing the scalar products $(x,w)_X = (y,z)_Y + (u,v)_U$ and $(p,q)_Q$ on $X_h$ and $Q_h$, we obtain

$$\underline{X}_h = \begin{pmatrix} \underline{Y}_h & 0\\ 0 & \nu\,M_h\end{pmatrix} \quad\text{and}\quad \underline{Q}_h = \frac{1}{\nu}\underline{Y}_h$$

with

$$\underline{Y}_h = \sqrt{\nu}\,K_h + M_h.$$

Observe that $\underline{Y}_h$ is the stiffness matrix representing the bilinear form $\sqrt{\nu}(\nabla y, \nabla q)_{L^2(\Omega)} + (\sqrt{\nu}+1)\,(y,q)_{L^2(\Omega)}$ on $Y_h$, which is of the same type as the bilinear form (on $Y$) of the state equation, but with modified coefficients.

It is easy to see that Lemma 4.1 remains valid with the same constants if $Y$, $U$, $Q$ are replaced by the finite-dimensional spaces $Y_h$, $U_h$, $Q_h$, as long as $Y_h = Q_h \subset U_h$.

As discussed before, it is reasonable to use a (properly scaled) preconditioner for $\underline{X}_h$ to approximate $\hat{A}_h$ and to use a (properly scaled) preconditioner for $\underline{Q}_h$ to approximate $\hat{S}_h$. For $\underline{Y}_h$, which appears in the first diagonal block of $\underline{X}_h$ and in $\underline{Q}_h$, we use, e.g., a standard multigrid preconditioner $\hat{Y}_h$ for the second order elliptic differential operator represented by the bilinear form $\sqrt{\nu}\,(\nabla y, \nabla q)_{L^2(\Omega)} + (\sqrt{\nu}+1)\,(y,q)_{L^2(\Omega)}$. For the well-conditioned matrix $M_h$, which appears in the second diagonal block of $\underline{X}_h$, a simple preconditioner $\hat{M}_h$, e.g., a few steps of a symmetric Gauss–Seidel iteration, is used. So, eventually we set

$$\hat{A}_h = \frac{1}{\sigma}\,\hat{X}_h = \frac{1}{\sigma}\begin{pmatrix}\hat{Y}_h & 0\\ 0 & \nu\,\hat{M}_h\end{pmatrix} \quad\text{and}\quad \hat{S}_h = \frac{\sigma}{\tau}\,\frac{1}{\nu}\,\hat{Y}_h \tag{4.1}$$

with real parameters $\sigma > 0$ and $\tau > 0$.

In summary, the preconditioner

$$\hat{\mathcal{K}}_h = \begin{pmatrix}\hat{A}_h & B_h^T\\ B_h & B_h\hat{A}_h^{-1}B_h^T - \hat{S}_h\end{pmatrix}$$

for the matrix

$$\mathcal{K}_h = \begin{pmatrix}A_h & B_h^T\\ B_h & 0\end{pmatrix}$$

is given by (4.1), where $\hat{Y}_h$ is a preconditioner for the second order elliptic differential operator represented by the bilinear form $\sqrt{\nu}\,(\nabla y, \nabla q)_{L^2(\Omega)} + (\sqrt{\nu}+1)\,(y,q)_{L^2(\Omega)}$ and a simple preconditioner $\hat{M}_h$ for the mass matrix.

It is reasonable to assume that

$$(1-q_X)\,\hat{Y}_h \le \underline{Y}_h \le \hat{Y}_h \quad\text{and}\quad (1-q_X)\,\hat{M}_h \le M_h \le \hat{M}_h$$

for some small value $q_X \in [0,1)$. The factor $q_X$ describes the quality of the preconditioners $\hat{Y}_h$ and $\hat{M}_h$.

The discussion in the previous section shows that the conditions (2.7) and (2.8) are satisfied with

$$\alpha = \sigma\,(1-q_X)\,\frac{2}{3} \quad\text{and}\quad \beta = \tau$$

for parameters $\sigma$ and $\tau$ satisfying

$$\sigma < 1 \quad\text{and}\quad \tau > \frac{4}{3(1-q_X)^2}.$$

In particular, assuming that $q_X \approx 0$, we can expect $\alpha \approx 2/3$ and $\beta \approx 4/3$ for $\sigma \approx 1$ and $\tau \approx 4/3$, leading to a rough estimate of the condition number $\kappa \approx \kappa(2/3, 4/3) \approx 4$, which implies a convergence factor $q \approx 1/3$ for the CG method.

## 5. Implementation issues.

The proposed method in this paper is the standard CG method applied to the preconditioned system

$$\hat{\mathcal{K}}_h^{-1}\mathcal{K}_h \begin{pmatrix} \underline{x}_h \\ \underline{p}_h \end{pmatrix} = \hat{\mathcal{K}}_h^{-1} \begin{pmatrix} \underline{f}_h \\ \underline{g}_h \end{pmatrix}$$

with the nonstandard scalar product

$$\left( \begin{pmatrix} \underline{x}_h \\ \underline{p}_h \end{pmatrix}, \begin{pmatrix} \underline{w}_h \\ \underline{q}_h \end{pmatrix} \right)_{\mathcal{D}_h} = ((\hat{A}_h - A_h)\underline{x}_h, \underline{w}_h) + ((B_h \hat{A}_h^{-1} B_h^T - \hat{S}_h)\underline{p}_h, \underline{q}_h).$$

For the matrices $\hat{A}_h$ and $\hat{S}_h$, preconditioners $\hat{X}_h$ and $\hat{Q}_h$ are needed which approximate the matrices $\underline{X}_h$ and $\underline{Q}_h$ representing the scalar products on the discrete spaces $X_h$ and $Q_h$, respectively. The discrete spaces $X_h$ and $Q_h$ typically involve discretizations of Sobolev spaces, whose scalar products are the bilinear forms associated with elliptic differential operators. So, in the end, good preconditioners for these elliptic differential operators are required, such as multilevel or multigrid preconditioners.

A straightforward implementation of the CG method would require the evaluation of the nonstandard scalar product, which can be done if the operation

$$\mathcal{D}_h \begin{pmatrix} \underline{w}_h \\ \underline{q}_h \end{pmatrix} \quad \text{with } \mathcal{D}_h = \begin{pmatrix} \hat{A}_h - A_h & 0 \\ 0 & B_h \hat{A}_h^{-1} B_h^T - \hat{S}_h \end{pmatrix} = \hat{\mathcal{K}}_h - \mathcal{K}_h$$

is available. This would involve matrix-vector products with the preconditioners $\hat{A}_h$ and $\hat{S}_h$, which is, in general, prohibitively costly for multilevel or multigrid preconditioners $\hat{A}_h$ and $\hat{S}_h$. A closer look at the CG method reveals that this operation is only required for vectors of the form

$$\begin{pmatrix} \underline{w}_h \\ \underline{q}_h \end{pmatrix} = \hat{\mathcal{K}}_h^{-1} \begin{pmatrix} \underline{s}_h \\ \underline{t}_h \end{pmatrix}.$$

But then

$$\mathcal{D}_h \begin{pmatrix} \underline{w}_h \\ \underline{q}_h \end{pmatrix} = \hat{\mathcal{D}}_h \hat{\mathcal{K}}_h^{-1} \begin{pmatrix} \underline{s}_h \\ \underline{t}_h \end{pmatrix} = (\hat{\mathcal{K}}_h - \mathcal{K}_h)\hat{\mathcal{K}}_h^{-1} \begin{pmatrix} \underline{s}_h \\ \underline{t}_h \end{pmatrix} = \begin{pmatrix} \underline{s}_h \\ \underline{t}_h \end{pmatrix} - \mathcal{K}_h \begin{pmatrix} \underline{w}_h \\ \underline{q}_h \end{pmatrix},$$

which shows that direct matrix-vector products with the preconditioners $\hat{A}$ and $\hat{S}_h$ are not needed. As discussed in section 2, the operation

$$\hat{\mathcal{K}}_h^{-1} \begin{pmatrix} \underline{s}_h \\ \underline{t}_h \end{pmatrix}$$

requires only operations of the form $\hat{A}_h^{-1}\tilde{\underline{s}}_h$ and $\hat{S}_h\tilde{\underline{t}}_h$, which are, of course, available for multilevel or multigrid preconditioners.

## 6. Numerical experiments.

We consider the optimal control problem from the previous section on the unit cube $\Omega = (0,1)^3$ and with homogeneous data $y_d \equiv 0$. Starting from an initial mesh of 24 tetrahedra (starting level $l = 1$), we obtain a hierarchy of nested meshes by uniform refinement up to some final level $l = L$. On each tetrahedral mesh, piecewise linear and continuous finite elements are used for $Y_h = U_h = Q_h$.

The discretized mixed problem is solved on the finest mesh (level $l = L$) by using the CG method for the preconditioned system (2.9) with the scalar product

(2.3) as described before. For the preconditioner we used the proposed symmetric block preconditioner, where $\hat{Y}_h$ is one $V$-cycle of the multigrid method with $m_1$ forward Gauss–Seidel steps for presmoothing and $m_1$ backward Gauss–Seidel steps for postsmoothing (in short $V(m_1, m_1)$) for the second order elliptic differential operator represented by the bilinear form $\sqrt{\nu}\,(\nabla y, \nabla q)_{L^2(\Omega)} + (\sqrt{\nu}+1)\,(y, q)_{L^2(\Omega)}$. For $\hat{M}_h$ we use $m_2$ steps of the symmetric Gauss–Seidel method (in short $SGS(m_2)$).

Starting values $\underline{x}_h^{(0)}$ and $\underline{p}_h^{(0)}$ are generated randomly. The exact solution of the problem is the trivial solution $\underline{x}_h = 0$ and $\underline{p}_h = 0$. The quality of an approximation $(\underline{x}_h^{(k)}, \underline{p}_h^{(k)})$ is measured by either the energy norm $e^{(k)}$ of the error, which here is given by

$$e^{(k)} = \left\| \begin{pmatrix} \underline{x}_h^{(k)} \\ \underline{p}_h^{(k)} \end{pmatrix} \right\|_{\mathcal{D}_h \hat{\mathcal{K}}_h^{-1} \mathcal{K}_h},$$

or the residual $r^{(k)}$:

$$r^{(k)} = \left\| \mathcal{K}_h \begin{pmatrix} \underline{x}_h^{(k)} \\ \underline{p}_h^{(k)} \end{pmatrix} \right\|.$$

All computations were performed on a Linux-PC with a 2.0GHz 64-bit processor and 3GB memory.

Figure 6.1 shows a typical convergence history (number of iterations $k$ versus $e^{(k)}/e^{(0)}$ and $r^{(k)}/r^{(0)}$) for level $L = 5$ (number of unknowns $3 \times 17{,}985$) and regularization parameter $\nu = 1$ using a $V(3,3)$-cycle for $\hat{Y}_h$ and $SGS(3)$ for $\hat{M}_h$ and parameters $\sigma = 0.9$ and $\tau = 1.1/k_0^2$ with $k_0^2 = 3/4$. The solid straight line with the circular markers illustrates the theoretically predicted behavior (convergence factor $q = 1/3$; see the discussion at the end of section 4), which is in good agreement with the observed behavior.

[illegible] 1. The convergence rate of the proposed method was shown to be bounded below 1 independently of $\nu$ (and $h$). However, the norm itself depends on $\nu$. This might lead to the suspicion that, nevertheless, the performance depends on the parameter $\nu$. Observe that the Euclidean norm of the residuals shows a similar behavior as the energy norm, which is not predicted by the theory. So, after a fixed number of iterations (here 30 iterations), the values of the residuals cannot be distinguished from 0 relative to the initial residual within machine precision. In this sense the numerical experiments confirm that the method is really robust in $\nu$.

Table 6.1 shows that the number of iterations does not depend on the level of refinement. $L$ denotes the level of refinement, $n+m$ the total number of all unknowns $\underline{y}_h$, $\underline{u}_h$, and $\underline{p}_h$, $k$ the number of iterations needed to satisfy the stopping rule

$$r^{(k)} \le \varepsilon\, r^{(0)} \quad \text{with } \varepsilon = 10^{-8},$$

and $t$ the total CPU time in seconds.

Table 6.2 shows that the number of iterations does not depend on the regularization parameter $\nu$ either. The results are given for refinement level $L = 5$.

**7. Concluding remarks.** Comparing the matrix $\mathcal{K}_h$ and the preconditioner $\hat{\mathcal{K}}_h$, a first remarkable observation is that the mass matrix $M_h$ (representing the $L^2$ inner product on $Y_h$) in the first diagonal block of $A_h$ is preconditioned by a preconditioner for a second order elliptic differential operator. Of course, such a preconditioner cannot be a good preconditioner for $M_h$ on the whole space $Y_h$, but it is a good

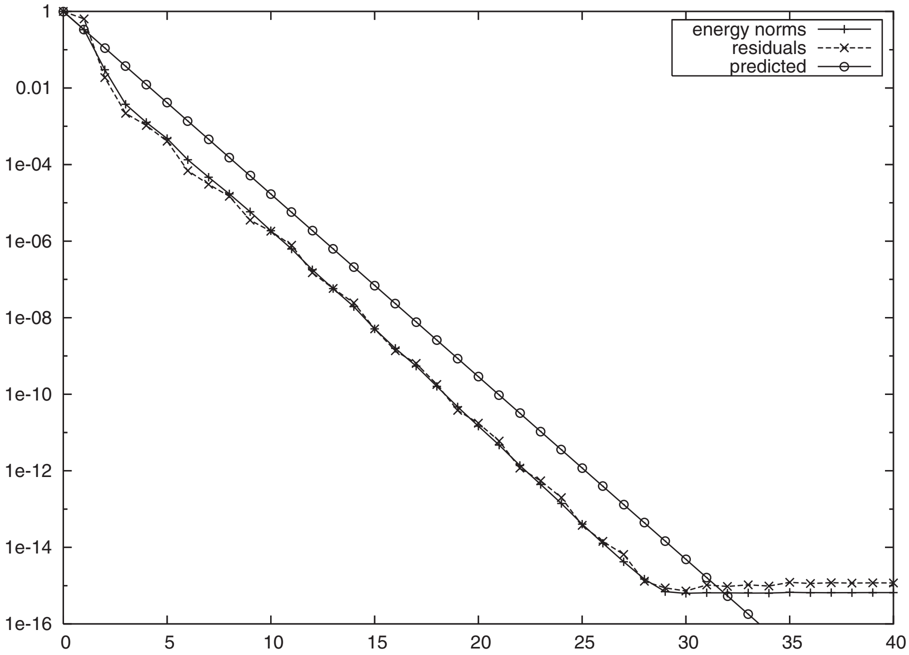


Fig. 6.1. *Convergence history: Number of iterations versus relative accuracy.*

Table 6.1
*Dependence of the number of iterations on the mesh size for fixed $\nu = 1$.*

| Level $L$ | Number of unknowns $n + m$ | Iterations $k$ | CPU time $t$ (in seconds) |
|---|---|---|---|
| 3 | 1,107 | 14 | 0.06 |
| 4 | 7,395 | 15 | 0.61 |
| 5 | 53,955 | 15 | 6.96 |
| 6 | 412,035 | 16 | 62.04 |
| 7 | 3,200,227 | 15 | 559.16 |

Table 6.2
*Dependence of the number of iterations on $\nu$ for fixed refinement level $L = 5$.*

| $\nu$ | Iterations $k$ |
|---|---|
| $10^{-4}$ | 15 |
| $10^{-2}$ | 14 |
| 1 | 15 |
| $10^{2}$ | 14 |
| $10^{4}$ | 15 |

preconditioner on the kernel of $B_h$, as it was shown. This suffices for the convergence analysis.

A more straightforward alternative would be to use some lumped mass matrix for preconditioning $M_h$ or even to use $M_h$ itself because it is well conditioned and, therefore, easy to invert. However, the resulting inexact Schur can then be interpreted as a discretized fourth order elliptic differential operator, for which it is much harder to find an efficient preconditioner. With our choice of the preconditioner for the mass matrix, the inexact Schur complement remains a discretized second order differential operator of the same complexity as the discretized second order differential operator of the state equation, for which an efficient preconditioner is usually available.

So in this context, it pays to invest (a little) more in preconditioning the mass matrix by a (properly scaled) Laplace-type preconditioner instead of some simple preconditioner. This would normally be considered a very obscure strategy. However, it is a very natural thing to do here because it just reflects the standard conditions of Brezzi's theorem.

A second remarkable observation concerns the discussed problem from optimal control. For the considered case of distributed control, it was shown theoretically and confirmed experimentally that the proposed preconditioner leads to convergence rates not only robust with respect to the mesh size $h$ but also robust with respect to the regularization parameter $\nu$.

**Acknowledgments.** We would like to thank the anonynmous referees for their valuable comments and suggestions which helped to improve this manuscript.

## REFERENCES

[1] G. Al-Jeiroudi, J. Gondzio, and J. Hall, *Preconditioning Indefinite Systems in Interior Point Methods for Large Scale Linear Optimization*, Technical report MS-2006-003, School of Mathematics, The University of Edinburgh, Edinburgh, Scotland, 2006.

[2] K. Arrow, L. Hurwicz, and H. Uzawa, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[3] R. E. Bank, B. D. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1990), pp. 645–666.

[4] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[5] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numer. Math., 103 (2006), pp. 173–196.

[6] J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17.

[7] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[8] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.

[9] H. S. Dollar, *Iterative Linear Algebra for Constrained Optimization*, Ph.D. Thesis, University of Oxford, Oxford, UK, 2005.

[10] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. Wathen, *Implicit-factorization preconditioning and iterative solvers for regularized saddle-point systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 170–189.

[11] N. Dyn and W. E. Ferguson, *The numerical solution of equality constrained quadratic programming problems*, Math. Comp., 41 (1983), pp. 165–170.

[12] H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[13] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, *Minimum residual methods for augmented systems*, BIT, 38 (1998), pp. 527–543.

[14] R. Fletcher, *Practical Methods of Optimization. Vol.* 2: *Constrained Optimization*, John Wiley & Sons, Chichester, UK, 1981.

[15] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary Value Problems*, North–Holland, Amsterdam, 1983.

[16] G. H. Golub and C. Greif, *On solving block-structured indefinite linear systems*, SIAM J. Sci. Comput., 24 (2003), pp. 2076–2092.

[17] G. H. Golub, C. Greif, and J. M. Varah, *An algebraic analysis of a block diagonal preconditioner for saddle point problems*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 779–792.

[18] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.

[19] W. Hackbusch, *Iterative Solutions of Large Sparse Systems of Equations*, Springer-Verlag, New York, 1994.

[20] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[21] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 368–391.

[22] T. Rusten and R. Winther, *A preconditioned iterative method for saddlepoint problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.

[23] Y. Saad and H. A. van der Vorst, *Iterative solution of linear systems in the* 20*th century*, J. Comput. Appl. Math., 123 (2000), pp. 1–33.

[24] D. Silvester and A. Wathen, *Fast iterative solution of stabilized Stokes systems. Part* II: *Using block diagonal preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.

[25] F. Tröltzsch, *Optimale Steuerung partieller Differentialgleichungen. Theorie, Verfahren und Anwendungen*, Vieweg, Wiesbaden, Germany, 2005.

[26] P. S. Vassilevski and R. D. Lazarov, *Preconditioning mixed finite element saddle-point elliptic problems*, Numer. Linear Algebra Appl., 3 (1996), pp. 1–20.

[27] W. Zulehner, *Analysis of iterative methods for saddle point problems: A unified approach*, Math. Comp., 71 (2002), pp. 479–505.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 3, pp. 774–795

# A HYBRID APPROACH COMBINING CHEBYSHEV FILTER AND CONJUGATE GRADIENT FOR SOLVING LINEAR SYSTEMS WITH MULTIPLE RIGHT-HAND SIDES*

GENE H. GOLUB[†], DANIEL RUIZ[‡], AND AHMED TOUHAMI[§]

**Abstract.** One of the most powerful iterative schemes for solving symmetric, positive definite linear systems is the conjugate gradient algorithm of Hestenes and Stiefel [*J. Res. Nat. Bur. Standards*, 49 (1952), pp. 409–435], especially when it is combined with preconditioning (cf. [P. Concus, G.H. Golub, and D.P. O'Leary, in *Proceedings of the Symposium on Sparse Matrix Computations*, Argonne National Laboratory, 1975, Academic, New York, 1976]). In many applications, the solution of a sequence of equations with the same coefficient matrix is required. We propose an approach based on a combination of the conjugate gradient method with Chebyshev filtering polynomials, applied only to a part of the spectrum of the coefficient matrix, as preconditioners that target some specific convergence properties of the conjugate gradient method. We show that our preconditioner puts a large number of eigenvalues near one and do not degrade the distribution of the smallest ones. This procedure enables us to construct a lower dimensional Krylov basis that is very rich with respect to the smallest eigenvalues and associated eigenvectors. A major benefit of our method is that this information can then be exploited in a straightforward way to solve sequences of systems with little extra work. We illustrate the performance of our method through numerical experiments on a set of linear systems.

**Key words.** linear systems, symmetric positive definite matrices, conjugate gradient algorithm, Chebyshev filtering polynomials, polynomial preconditioning, iterative methods

**AMS subject classifications.** 65F10, 65F15, 65F50, 65H10, 65H17, 65N22

**DOI.** 10.1137/060649458

## 1. Introduction.

The preconditioned conjugate gradient algorithm is among the most powerful techniques for solving linear systems of the form

$$\mathbf{A}\mathbf{x}^\star = \mathbf{b}, \tag{1.1}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is large and symmetric positive definite, and $\mathbf{b} \in \mathbb{R}^n$ is a given right-hand side (see, e.g., [8]). Concerning preconditioning, the use of polynomial preconditioners is one attractive possibility, considered by several authors [18, 24]. The polynomial preconditioner is generally chosen so that the preconditioned matrix has an eigenvalue distribution that is favorable to the conjugate gradient method, i.e., either with a largely reduced condition number so that the conjugate gradient method applied to the preconditioned system converges rapidly and/or with eigenvalues confined to a small number of intervals.

---

[*] Received by the editors January 9, 2006; accepted for publication (in revised form) by D. B. Szyld February 21, 2007; published electronically July 11, 2007.
http://www.siam.org/journals/simax/29-3/64945.html

[†] Department of Computer Science, SC-CM Program, Stanford University, Gates 2B, Stanford, CA 94305-9025 (golub@stanford.edu). The work of this author was supported in part by the DOE.

[‡] ENSEEIHT-IRIT, 2 rue C. Camichel, 31071 Toulouse Cedex, France (Daniel.Ruiz@enseeint.fr). The work of this author was partially supported by the ACI GRID TLSE project from the French ministry of research and education, grant 02 2 0678.

[§] Corresponding author. ENSEEIHT-IRIT, 2 rue C. Camichel, 31071 Toulouse Cedex, France. Current address: Hassan 1st University, Faculty of Sciences and Technologies, Mathematics and Computer Science Department, BP: 577, route de Casablanca, Settat, Morocco (Ahmed.Touhami@fsts.ac.ma, Ahmed.Touhami@uh1.ac.ma, Ahmed.Touhami@enseeiht.fr). This author's work was partially supported by the ACI GRID TLSE project from the French ministry of research and education, grant 02 2 0678.

For the Chebyshev iteration, estimates of $\lambda_{\min}$ and $\lambda_{\max}$ are needed. Often, good upper bounds $\lambda_{\max}$ can be obtained easily, using simple techniques such as Gershgorin's theorem [24]. It is far more difficult to estimate the smallest eigenvalue. Saad [24] proposed a polynomial preconditioning technique that requires only an upper bound for the largest eigenvalue, while the trivial bound $\lambda_{\min} = 0$ is used for the smallest eigenvalue of $\mathbf{A}$. His technique is based on the least-squares polynomials associated with the family of Jacobi weights [28]. Another way to avoid the computation of $\lambda_{\min}$ and $\lambda_{\max}$ is to use the conjugate gradient polynomial itself as a preconditioner. This approach was investigated by O'Leary [22]. The disadvantage of this technique is that the preconditioned system is not guaranteed to be positive definite. Golub and Kent [14] gave a method for estimating the extreme eigenvalues based on modified moments. Ashby, Manteuffel, and Otto [3] demonstrated in a variety of numerical examples that the effectiveness of Chebyshev and least-squares polynomial preconditioners depends on the eigenvalue distribution of the coefficient matrix $\mathbf{A}$.

Here, a different approach is proposed that aims at putting the conjugate gradient in [illegible] mode, where the condition number of the preconditioned matrix is not so much reduced but its spectrum is largely clustered around one. Our preconditioner, called the ChebFilter, consists of applying the Chebyshev filtering polynomials only to a part of the spectrum of the given iteration matrix in an attempt to shift the maximum number of eigenvalues of the coefficient matrix close to one without degrading the distribution of the smallest eigenvalues. With this preconditioning technique, the conjugate gradient method still exhibits plateaus in its convergence behavior and does not enter linear mode as usually expected. These plateaus, which are mostly due to the combination of ill-conditioning and clusters of eigenvalues at the extremes, are greatly reduced in length, and the resulting method distinguishes itself by its ability to construct small dimensional Krylov bases that are very rich with respect to the smallest eigenvalues and associated eigenvectors.

In the following, we assume that $\mathbf{A}$ is a large s.p.d. matrix with a spectrum largely clustered, and the goal is to solve a sequence of large linear systems involving the same matrix but different right-hand sides. For simultaneous right-hand sides, block Krylov linear solvers [21] might be appropriate; for a sequence of right-hand sides that do not vary much, a straightforward idea is to use the former solution as an initial guess for the next solution. If only one right-hand side is available at a time, the method of Fischer [11], the deflated conjugate gradient method (deflated CG) [26], or the hybrid method of Simoncini and Gallopoulos [27] may be employed. Fischer's method first looks for a solution in the space spanned by the previous solution vectors in the sequence, which is helpful only if the solution vectors are correlated. In the deflated CG method, only a small number of the initial Lanczos vectors for every system are used to update the approximate invariant subspace. This is efficient, in both computation and memory use, but the convergence to an invariant subspace is slow. Hence, the improvement in the number of iterations is modest. The hybrid method of Simoncini and Gallopoulos is most effective only when the right-hand sides share common spectral information.

The alternative approach that we shall investigate in this work is not much different in spirit from these techniques but takes advantage of the particular Chebyshev preconditioners to construct a Krylov basis of small dimension very rich with respect to the eigeninformation linked to the smallest eigenvalues, without the need for any postprocessing. The main idea is to solve the first linear system with conjugate gradient combined with Chebyshev filters as a preconditioner and to exploit the resulting complete Krylov basis, generated at the first solution in the sequence, either to deflate the initial residual before using the classical conjugate gradient in the next solutions, or to build some particular preconditioned or deflated conjugate gradient algorithms. This approach is also rather independent of the right-hand sides, because the Chebyshev polynomial filters act uniformly on the eigencomponents independently of the residual vectors. The conjugate gradient, on the other hand, constructs a polynomial explicitly linked with these eigencomponents to minimize the **A**-norm of the current error vector.

The outline of the paper is as follows. Section 2 presents our preconditioner and discusses some of its properties. In section 3, we illustrate the efficiency of the proposed technique on a set of model problems arising from the discretization via finite elements of some 2D heterogeneous diffusion PDE problems. Section 4 shows how we can use the Chebyshev filter with the conjugate gradient to solve a sequence of linear systems involving the same matrix but different right-hand sides. In section 5, we analyze in more detail the computational complexity of our approach. This technique is better than the conjugate gradient only for the solution of a sequence of linear systems with the same matrix but changing right-hand sides. We analyze in particular how quickly the extra work at the beginning, when solving the first system in the sequence, can be paid back after a few consecutive solutions. Finally, we conclude with some remarks and perspectives in section 6.

**2. Chebyshev filtering polynomials as a preconditioner.** The present section is devoted to the description of the proposed combination of conjugate gradients with Chebyshev filtering polynomials as preconditioners called ChebFilterCG. To describe this in detail, we first introduce the eigendecomposition of the s.p.d. matrix **A**:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{U}_1^T + \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^T,$$

where the spectrum of **A** is split in two parts: $\mathbf{\Lambda}_1$ is the diagonal matrix containing all eigenvalues of **A** less than a given positive number $\mu$, where $0 < \mu < \lambda_{\max}$ is fixed (user-given) and denotes the [illegible]. $\mathbf{U}_1$ is the rectangular matrix whose columns are the corresponding orthonormal set of eigenvectors in matrix form, and $\mathbf{\Lambda}_2$ and $\mathbf{U}_2$ are the corresponding complementary matrices.

Let $\mathbf{x}^{(0)} \in \mathbb{R}^n$ be any initial guess for the solution of (1.1), and let $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$ be the associated residual vector. We then introduce the [illegible] vector

$$\mathbf{w}_f = \mathcal{F}_m(\mathbf{A})\,\mathbf{r}^{(0)} = \mathbf{U}_1\mathcal{F}_m(\mathbf{\Lambda}_1)\mathbf{U}_1^T\,\mathbf{r}^{(0)} + \mathbf{U}_2\mathcal{F}_m(\mathbf{\Lambda}_2)\mathbf{U}_2^T\,\mathbf{r}^{(0)}, \tag{2.1}$$

where $\mathcal{F}_m$ is a polynomial function of degree $m$ given by

$$\mathcal{F}_m(\lambda) = \frac{T_m(\Theta_\mu(\lambda))}{T_m(\Theta_\mu(0))}, \tag{2.2}$$

with $T_m$ being the usual Chebyshev polynomial of degree $m$, and $\Theta_\mu$ is the linear mapping function that maps the interval $[\mu, \lambda_{\max}]$ onto the unit interval $[-1, 1]$ (with $\Theta_\mu(\mu) = 1$ and $\Theta_\mu(\lambda_{\max}) = -1$).

For given values of $\mu$, $\lambda_{\max}(\mathbf{A})$, and $\varepsilon$, we can fix the degree $m$ of $T_m$ such that $1/|T_m(\Theta_\mu(0))| < \varepsilon$ and consequently $||\mathcal{F}_m(\lambda)||_\infty < \varepsilon$ on $[\mu, \lambda_{\max}]$. Using (2.1), we can then write

$$||\mathbf{U}_2^T\,\mathbf{w}_f||_2 \le ||\mathcal{F}_m(\mathbf{\Lambda}_2)||_2\,||\mathbf{U}_2^T\,\mathbf{r}^{(0)}||_2 \le \varepsilon\,||\mathbf{U}_2^T\,\mathbf{r}^{(0)}||_2\,. \tag{2.3}$$

Equation (2.3) explains explicitly how the action of the Chebyshev filtering in **A** applied on a given vector $\mathbf{r}^{(0)}$ can reduce, below a value $\varepsilon$, the eigencomponents in

ALGORITHM 2.1: CHEBFILTER

```
$[\mathbf{w}_f, \mathbf{x}_f] = \text{ChebFilter}\,\big(\mathbf{A}, \mathbf{b}, \mu, \lambda_{\max}(\mathbf{A}), \mathbf{x}^{(0)}, \varepsilon\big)$

Begin
  1. $\alpha_\mu = \frac{2}{\lambda_{\max}(\mathbf{A}) - \mu}$ and $d_\mu = \frac{\lambda_{\max}(\mathbf{A}) + \mu}{\lambda_{\max}(\mathbf{A}) - \mu}$
  2. Given the initial guess $\mathbf{x}^{(0)}$
  3. $\mathbf{x}_f = \mathbf{x}^{(0)}$, set $\mathbf{w}_f = \mathbf{b} - \mathbf{A}\mathbf{x}_f$
  4. $\mathbf{y} = \mathbf{x}_f,\ m = 1,\ \sigma_0 = 1$, and $\sigma_1 = d_\mu$
  5. $\mathbf{x}_f = \mathbf{x}_f + \frac{\alpha_\mu}{d_\mu}\mathbf{w}_f$ and $\mathbf{w}_f = \mathbf{b} - \mathbf{A}\mathbf{x}_f$
  6. Do While $1/\sigma_m \geq \varepsilon$
      i. $\sigma_{m+1} = 2\, d_\mu\, \sigma_m - \sigma_{m-1}$
      ii. $\mathbf{p} = 2\,\frac{\sigma_m}{\sigma_{m+1}}\Big(d_\mu \mathbf{x}_f + \alpha_\mu \mathbf{w}_f\Big) - \frac{\sigma_{m-1}}{\sigma_{m+1}}\,\mathbf{y}$
      iii. $\mathbf{y} = \mathbf{x}_f$ and $\mathbf{x}_f = \mathbf{p}$
      iv. $\mathbf{w}_f = \mathbf{b} - \mathbf{A}\mathbf{x}_f$
      v. $m = m + 1$
  7. EndDo
End
```

$\mathbf{r}^{(0)}$ associated to all eigenvalues in the range $[\mu, \lambda_{\max}(\mathbf{A})]$ relative to the others. The number of Chebyshev steps required to achieve a given level of filtering $\varepsilon$ is directly related to the rate of convergence of Chebyshev polynomials on the interval $[\mu, \lambda_{\max}]$ (see, e.g., [14], [16, p. 47]), which depends only on the ratio $\lambda_{\max}/\mu$.

We can then introduce the Algorithm 2.1, which we call ChebFilter and which corresponds to the application of a Chebyshev polynomial in $\mathbf{A}$ to reduce by a factor $\varepsilon$ the eigencomponents in $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$ associated with all eigenvalues in the range $[\mu, \lambda_{\max}(\mathbf{A})]$. As a result, the algorithm provides the filtered residual $\mathbf{w}_f = \mathcal{F}_{m+1}(\mathbf{A})\,\mathbf{r}^{(0)}$ and the corresponding iterate approximation $\mathbf{x}_f$ such that $\mathbf{b} - \mathbf{A}\mathbf{x}_f = \mathbf{w}_f$. In this algorithm, steps 6.ii and 6.iii are connected together with the following relation:

$$\mathbf{w}_f = \mathbf{w}^{(m+1)} = \mathcal{F}_{m+1}(\mathbf{A})\,\mathbf{r}^{(0)} = \frac{T_{m+1}(\Theta_\mu(\mathbf{A}))\,\mathbf{r}^{(0)}}{T_{m+1}(d_\mu)} = \frac{1}{\sigma_{m+1}}\,T_{m+1}\Big(d_\mu\,\mathbf{I} - \alpha_\mu\,\mathbf{A}\Big)\,\mathbf{r}^{(0)},$$

where we note $d_\mu = \frac{\lambda_{\max}+\mu}{\lambda_{\max}-\mu}$, $\alpha_\mu = \frac{2}{\lambda_{\max}-\mu}$, and $\sigma_m = T_m(d_\mu)$ for all $m \geq 0$. In that respect, step 6.iv can also be replaced by the equivalent following three-term recurrence relation:

$$\mathbf{w}_f = \mathbf{w}^{(m+1)} = 2\,\frac{\sigma_m}{\sigma_{m+1}}\Big(d_\mu \mathbf{w}^{(m)} - \alpha_\mu \mathbf{A}\mathbf{w}^{(m)}\Big) - \frac{\sigma_{m-1}}{\sigma_{m+1}}\,\mathbf{w}^{(m-1)}, \tag{2.4}$$

which corresponds to the usual Chebyshev three-term recurrence formula giving $\mathbf{w}^{(m+1)}$ in a function of $\mathbf{w}^{(m)}$ and $\mathbf{w}^{(m-1)}$ for $m \geq 1$, with $\mathbf{w}^{(0)}$ set to $\mathbf{r}^{(0)}$ at the beginning and $\mathbf{w}^{(1)}$ set to $\big(\mathbf{I} - \frac{\alpha_\mu}{d_\mu}\mathbf{A}\big)\mathbf{r}^{(0)}$.

Let us denote now by $\mathbf{x}^{(i)}$ the $i$th iterate in the preconditioned conjugate gradient and by $\mathbf{r}^{(i)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(i)} = \mathbf{A}(\mathbf{x} - \mathbf{x}^{(i)})$ the associated residual. The application of the ChebFilter polynomial as a preconditioner consists in solving approximatively the linear system $\mathbf{A}\mathbf{z}^{(i)} = \mathbf{r}^{(i)}$ such that its final residual is given by $\mathbf{r}^{(i)} - \mathbf{A}\mathbf{z}^{(i)} = \mathcal{F}_m(\mathbf{A})\,\mathbf{r}^{(i)}$, which can be expressed as

$$\mathbf{z}^{(i)} = \mathbf{A}^{-1}\big(\mathbf{I} - \mathcal{F}_m(\mathbf{A})\big)\mathbf{r}^{(i)} = \mathbf{M}^{-1}\mathbf{r}^{(i)}. \tag{2.5}$$

We mention that our polynomial preconditioner, in the way it is implemented, carries out a fixed number of steps for given values of $\lambda_{\max}(\mathbf{A})$, $\mu$, and $\varepsilon$, independently of the right-hand side.

The $i$th iteration in the preconditioned conjugate gradient consists of searching $\mathbf{x}^{(i)} \in \{\mathbf{x}^{(0)}\} + \mathcal{K}_i(\mathbf{M}^{-1}\mathbf{A}, \mathbf{z}^{(0)})$, with $\mathbf{z}^{(0)} = \mathbf{M}^{-1}\mathbf{b}$ and $\mathbf{r}^{(i)} \perp \mathcal{K}_i(\mathbf{M}^{-1}\mathbf{A}, \mathbf{z}^{(0)})$. $\mathcal{K}_i(\mathbf{M}^{-1}\mathbf{A}, \mathbf{z}^{(0)})$ is the $i$th Krylov subspace and is given by

$$\begin{aligned}\mathcal{K}_i\big(\mathbf{M}^{-1}\mathbf{A}, \mathbf{z}^{(0)}\big) &= \mathrm{Span}\Big(\mathbf{z}^{(0)}, \dots, \big(\mathbf{M}^{-1}\mathbf{A}\big)^{\mathrm{i}-1}\mathbf{z}^{(0)}\Big)\\ &= \mathrm{Span}\Big(\mathbf{z}^{(0)}, \dots, \big(\mathbf{I} - \mathcal{F}_{\mathrm{m}}(\mathbf{A})\big)^{\mathrm{i}-1}\mathbf{z}^{(0)}\Big)\\ &= \mathrm{Span}\Big(\mathbf{z}^{(0)}, \dots, \big(\mathcal{F}_{\mathrm{m}}(\mathbf{A})\big)^{\mathrm{i}-1}\mathbf{z}^{(0)}\Big),\end{aligned} \tag{2.6}$$

since $\mathbf{A}^{-1}\mathcal{F}_m(\mathbf{A}) = \mathcal{F}_m(\mathbf{A})\mathbf{A}^{-1}$, and consequently $\mathbf{M}^{-1}\mathbf{A} = \mathbf{A}^{-1}\big(\mathbf{I} - \mathcal{F}_m(\mathbf{A})\big)\mathbf{A} = \big(\mathbf{I} - \mathcal{F}_m(\mathbf{A})\big)\mathbf{A}^{-1}\mathbf{A}$. Equation (2.6) shows explicitly that the Chebyshev preconditioned conjugate gradient search directions remain in a filtered Krylov subspace.

Note that, since $\mathbf{A}$ is s.p.d., then $\mathbf{M}^{-1} = \mathbf{A}^{-1}\big(\mathbf{I} - \mathcal{F}_m(\mathbf{A})\big)$ is symmetric positive definite. Indeed, since $\mathbf{A}^{-1}$ and $\mathcal{F}_m(\mathbf{A})$ commute, it is easy to see that $\mathbf{M}^{-1}$ is symmetric. Finally, using the eigendecomposition of the s.p.d. matrix $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, we can write $\mathbf{M}^{-1} = \mathbf{U}\boldsymbol{\Lambda}^{-1}\big(\mathbf{I} - \mathcal{F}_m(\boldsymbol{\Lambda})\big)\mathbf{U}^T$, where $\boldsymbol{\Lambda}^{-1}\big(\mathbf{I} - \mathcal{F}_m(\boldsymbol{\Lambda})\big)$ is diagonal positive definite since $\mathcal{F}_m(\lambda) \in [-\varepsilon, 1[$ for all $\lambda$ in $]0, \lambda_{\max}]$. In addition we can see that the matrices $\mathbf{M}^{-1}\mathbf{A}$ and $\mathbf{A}\mathbf{M}^{-1}$ have the same eigenvalues, which are those of $(\mathbf{I} - \mathcal{F}_m(\mathbf{A}))$. Therefore, the eigenvalues of $\mathbf{M}^{-1}\mathbf{A}$ are given by $1 - \mathcal{F}_m(\lambda_i)$, with $\lambda_i$, $i \in \{1, \dots, n\}$, the eigenvalues of $\mathbf{A}$.

An important aspect in the use of such a Chebyshev filter as a preconditioner is that it acts uniformly on the range $[\mu, \lambda_{\max}(\mathbf{A})]$, independently of the right-hand side, as opposed to the conjugate gradient, which finds the polynomial minimizing the $\mathbf{A}$-norm of the error in the given linear system.

## 3. Numerical experiments.

In this section, we illustrate with some numerical experiments the efficiency of the polynomial preconditioner introduced above. Our test problems PDE1 and PDE2 are extracted from the finite element discretization using MATLAB of the partial differential equation problem:

$$\left\{\begin{array}{rcl} -\mathrm{div}\Big(\Lambda(x)\cdot\nabla u\Big) & = & f \quad \text{in} \quad \Omega,\\ u_{|\partial\Omega} & = & 0, \end{array}\right.$$

where $\Omega \subset \mathbb{R}^2$ is an L-shaped region as described in Figure 3.1.

The major differences between PDE1 and PDE2 are in the settings for $\Lambda(x)$ and in the size of discretization.

In the PDE1 problem, $f = 10$, and the function $\Lambda(x) \in L^\infty(\Omega)$ takes different scalar values in each subdomain:

$$\Lambda(x) = \left\{\begin{array}{ll} 1 & x \in \Omega_1 \cup \Omega_4,\\ 10^6 & x \in \Omega_2,\\ 10^4 & x \in \Omega_3. \end{array}\right.$$

The resulting linear system (1.1) has 7,969 degrees of freedom, the number of nonzero elements in the coefficient matrix $\mathbf{A}$ is $\mathit{nnz}\,(\mathbf{A}) = 55,131$, and the norm of $\mathbf{A}$ is equal to $9.54 \cdot 10^6$.

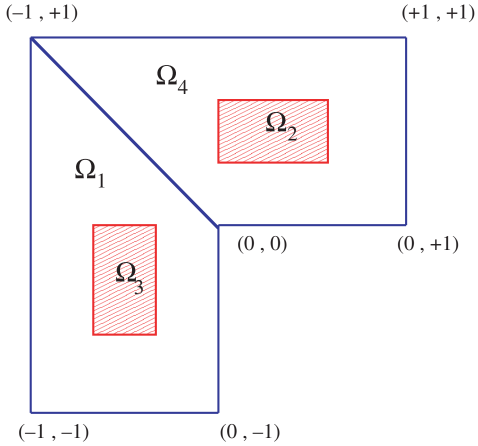


FIG. 3.1. *Geometry of the domain* $\Omega$.

In the PDE2 problem, $f = 200$, and the PDE problem incorporates some heterogeneity and anisotropy with the matrix $\Lambda(x)$ that takes the following values:

$$\Lambda(x) = \begin{cases} \begin{bmatrix} 1 & 4\,\lambda_1 \\ 4\,\lambda_1 & \lambda_1 \end{bmatrix} & \text{if } x \in \Omega_1 \quad \text{and} \quad \lambda_2\,\mathbf{I}_2 \quad \text{if } x \in \Omega_2, \\ \begin{bmatrix} \lambda_1 & -2\,\lambda_1 \\ -2\,\lambda_1 & 1 \end{bmatrix} & \text{if } x \in \Omega_4 \quad \text{and} \quad \lambda_3\,\mathbf{I}_2 \quad \text{if } x \in \Omega_3, \end{cases}$$

where $\lambda_1 = 6 \cdot 10^{-2}$, $\lambda_2 = 1 \cdot 10^6$, $\lambda_3 = 1 \cdot 10^2$, and $\mathbf{I}_2$ denotes the $(2 \times 2)$ identity matrix. The resulting linear system (1.1) has 161,313 degrees of freedom, the number of nonzero elements in the coefficient matrix $\mathbf{A}$ is $\mathit{nnz}\,(\mathbf{A}) = 1,125,897$, and the norm of $\mathbf{A}$ is equal to $1.00 \cdot 10^7$.

Moreover, we use three kinds of the first level of preconditioners: the classical Jacobi diagonal matrix $\mathbf{M}_1 = \text{diag}(\mathbf{A})$, the incomplete Cholesky decomposition of $\mathbf{A}$ with no fill-in, and the incomplete Cholesky decomposition of $\mathbf{A}$ with drop tolerance $10^{-2}$, $\mathsf{cholinc}(\mathbf{A}, 10^{-2})$ as in MATLAB notations.

Using the incomplete Cholesky decompositions, we compute the lower triangular matrix $\mathbf{L}$ such that $\mathbf{M}_1 = \mathbf{L}\mathbf{L}^T$. The purpose of this first level of preconditioner is to better cluster the spectrum of our iteration matrix, which is favorable to the solution technique we propose here. Additionally, in the different figures illustrating the experimental results, we shall denote by "Classical CG" the preconditioned conjugate gradient with the classical Jacobi or the incomplete Cholesky factorization as the first level of preconditioner only.

First, we consider the case of the PDE1 problem. In Table 3.1, we report on the values of the condition numbers $\kappa(\mathbf{A})$ and $\kappa(\mathbf{M}_1^{-1}\mathbf{A})$ for the Jacobi and the incomplete Cholesky with no fill-in and with drop tolerance $10^{-2}$. The condition number of $\mathbf{A}$ is of order $10^9$; after incomplete Cholesky preconditioning with drop tolerance $10^{-2}$, $\kappa(\mathbf{M}_1^{-1}\mathbf{A})$ is of order $10^6$.

In Table 3.2, we indicate the number of eigenvalues in the interval $[\lambda_{\min}, \mu]$ in the case of the PDE1 problem for a large range of the value of the parameter $\mu$ for each preconditioner. From these data, we can see that the original matrix is very badly scaled, and that the Jacobi preconditioner is able to cluster the eigenvalues quite substantially, although the condition number remains very large.

TABLE 3.1
*Estimates for $\kappa(\mathbf{M}_1^{-1}\mathbf{A})$, $\lambda_{\min}$, and $\lambda_{\max}$ for the PDE1 problem.*

| First level preconditioner | $\kappa(\mathbf{M}_1^{-1}\mathbf{A})$ | $\lambda_{\min}$ | $\lambda_{\max}$ |
|---|---|---|---|
| $\mathbf{M}_1 = \mathbf{I}$ | $2.6 \cdot 10^9$ | $3.7 \cdot 10^{-3}$ | $9.6 \cdot 10^6$ |
| $\mathbf{M}_1 =$ Classical Jacobi ; $\mathbf{M}_1 = \mathrm{diag}(\mathbf{A})$ | $6.8 \cdot 10^8$ | $3.1 \cdot 10^{-9}$ | 2.08 |
| $\mathbf{M}_1 =$ IC ; no fill-in | $9.4 \cdot 10^7$ | $1.7 \cdot 10^{-8}$ | 1.6 |
| $\mathbf{M}_1 =$ IC ; drop tolerance $10^{-2}$ | $6.2 \cdot 10^6$ | $1.8 \cdot 10^{-7}$ | 1.1 |

TABLE 3.2
*Number of eigenvalues in $[\lambda_{\min}, \mu]$ for the PDE1 problem.*

| $\mu = \lambda_{\max}/\gamma$ | Preconditioner $\mathbf{M}_1$ | | | |
|---|---|---|---|---|
| $\gamma$ | Identity | Jacobi scaling | Inc. Cholesky(0) | Inc. Cholesky($10^{-2}$) |
| $10^9$ | 3 | | | |
| $10^8$ | 41 | | | |
| $10^7$ | >200 | | | |
| $10^3$ | | 3 | | |
| 500 | | 5 | | |
| 200 | | 18 | | |
| 100 | | 43 | 3 | |
| 50 | | 89 | 11 | |
| 20 | | >200 | 32 | |
| 10 | | | 68 | 3 |
| 5 | | | 157 | 9 |
| 2 | | | >200 | 40 |

In the following tables, the CPU times will not be shown, since all of our experiments were performed using MATLAB, to illustrate the numerical features of the proposed approach.

**3.1. Impact of the cutoff value and level of filtering.** The CHEBFILTER preconditioner developed in section 2 depends on two different parameters, namely, the choice of the cutoff filtering value $\mu$ and the filtering level $\varepsilon$. In Table 3.3, we consider the case of the PDE1 problem; the initial guess is $\mathbf{x}^{(0)} = 0$, and the right-hand side is chosen so that the solution $\mathbf{x}^\star$ of the linear system is the vector of all ones. In this table, we display the number of Chebyshev filtering steps (`ChebIt`), the number of iterations of CG (`CGIt`), and the total number matrix-vector products (`ChebIt`×`CGIt`) for different values of the filtering level $\varepsilon$ and for the cutoff filtering value $\mu$. In these tests, we have stopped the CG iterations when $\|\mathbf{x}^\star - \mathbf{x}^{(k)}\|_{\mathbf{A}} \leq 10^{-9}\ \|\mathbf{x}^\star\|_{\mathbf{A}}$. From these data, we can observe the combined effects of these two parameters on our test examples.

The parameter $\mu$ splits the spectrum of the matrix $\mathbf{A}$ in two subsets and determines the convergence rate of the classical Chebyshev iterations that will be performed in each conjugate gradient iteration for preconditioning, since this defines the damping interval $[\mu, \lambda_{\max}]$ where the Chebyshev polynomial uniformly converges to 0. The rapid change in the Chebyshev rate of convergence with smaller values of $\mu$ induces many more Chebyshev filtering steps at each iteration, and this can be counterbalanced only by a very strong reduction in the total number of iterations in the preconditioned conjugate gradient algorithm. In other words, it is worth reducing the value of $\mu$ only if there is a very strong clustering of eigenvalues in the spectrum of the iteration matrix. In that respect, a first level of preconditioning $\mathbf{M}_1$ is a key issue that may help to enforce this situation in the iteration matrix $\mathbf{M}_1^{-1}\mathbf{A}$.

Table 3.3

*Comparison of the number of Chebyshev filtering steps, the number of iterations of CG, and the total number of matrix-vector products for different values of the filtering level and different bounds for the damping interval for the PDE1 problem.*

- `CGIt` *denotes the number of conjugate gradient iterations.*
- `ChebIt` *denotes the number of Chebyshev filtering steps for preconditioning at each CG iteration.*
- `ChebIt`×`CGIt` *denotes the total number of matrix-vector products.*

| | PDE1 preconditioned with Jacobi | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | CG without Chebyshev preconditioning performs `CGIt` = 485 | | | | | | | | |
| | $\mu = \lambda_{\max}/500$ | | | $\mu = \lambda_{\max}/100$ | | | $\mu = \lambda_{\max}/50$ | | |
| Value of $\varepsilon$ | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` |
| $10^{-16}$ | 420 | 5 | 2100 | 188 | 11 | 2068 | 132 | 16 | 2112 |
| $10^{-8}$ | 214 | 6 | 1284 | 96 | 16 | 1536 | 68 | 23 | 1564 |
| $10^{-4}$ | 111 | 8 | 888 | 50 | 22 | 1100 | 35 | 31 | 1085 |
| $10^{-2}$ | 60 | 12 | 720 | 27 | 30 | 810 | 19 | 43 | 817 |
| $10^{-1}$ | 34 | 19 | 646 | 15 | 42 | 630 | 11 | 59 | 649 |

| | PDE1 preconditioned with incomplete Cholesky (no fill-in) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | CG without Chebyshev preconditioning performs `CGIt` = 191 | | | | | | | | |
| | $\mu = \lambda_{\max}/100$ | | | $\mu = \lambda_{\max}/50$ | | | $\mu = \lambda_{\max}/20$ | | |
| Value of $\varepsilon$ | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` |
| $10^{-16}$ | 188 | 4 | 752 | 132 | 6 | 792 | 83 | 9 | 747 |
| $10^{-8}$ | 96 | 5 | 480 | 68 | 8 | 544 | 43 | 13 | 559 |
| $10^{-4}$ | 50 | 7 | 350 | 35 | 11 | 385 | 22 | 18 | 396 |
| $10^{-2}$ | 27 | 11 | 297 | 19 | 15 | 285 | 12 | 25 | 300 |
| $10^{-1}$ | 15 | 18 | 270 | 11 | 22 | 242 | 7 | 35 | 245 |

| | PDE1 preconditioned with incomplete Cholesky (tol=$10^{-2}$) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | CG without Chebyshev preconditioning performs `CGIt` = 54 | | | | | | | | |
| | $\mu = \lambda_{\max}/20$ | | | $\mu = \lambda_{\max}/10$ | | | $\mu = \lambda_{\max}/5$ | | |
| Value of $\varepsilon$ | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` | `ChebIt` | `CGIt` | `ChebIt`×`CGIt` |
| $10^{-16}$ | 83 | 4 | 332 | 58 | 4 | 232 | 39 | 5 | 195 |
| $10^{-8}$ | 43 | 4 | 172 | 30 | 5 | 150 | 20 | 7 | 140 |
| $10^{-4}$ | 22 | 6 | 132 | 16 | 7 | 102 | 11 | 9 | 99 |
| $10^{-2}$ | 12 | 10 | 120 | 9 | 10 | 90 | 6 | 13 | 78 |
| $10^{-1}$ | 7 | 15 | 105 | 5 | 16 | 80 | 4 | 17 | 68 |

The second parameter, the filtering level $\varepsilon$, directly influences the number of Chebyshev steps. Note that, for a fixed level of filtering $\varepsilon$, the total number of matrix-vector products (`ChebIt`×`CGIt`) does not change dramatically when varying $\mu$.

For a fixed value of $\mu$, if we consider the total number of matrix-vector products (`ChebIt`×`CGIt`), it is clear that the conjugate gradient alone performs better than the combination of conjugate gradient with Chebyshev filters, simply because the Chebyshev polynomial is not optimal in the **A**-norm and acts uniformly and independently of the eigencomponents of the given residual. We shall see, however, that if we intend to reuse the generated filtered Krylov basis for the acceleration of further solutions, the size of this basis is of great importance, and smaller values of the filtering level $\varepsilon$ should be considered to minimize that size.

If we consider a filtering value $\varepsilon$ close to $10^{-1}$ to minimize the effort within each Chebyshev filtering step, we can observe that the total number of matrix-vector

Table 3.4

*Relative error on the smallest eigenvalues $\lambda_i$, $1 \leq i \leq 7$, for a level of filtering $\varepsilon = 10^{-4}$ and for different cutoff filtering values $\mu$. The problem considered is PDE1.*

| Matrix preconditioned with IC($10^{-2}$) | | | Matrix preconditioned with IC(0) | | |
|---|---|---|---|---|---|
| $\lambda_{\max} = 1.1$ | Value of $\|\lambda_i - \delta_i\|/\|\lambda_i\|$ | | $\lambda_{\max} = 1.6$ | Value of $\|\lambda_i - \delta_i\|/\|\lambda_i\|$ | |
| $\lambda_i$ | $\mu = \lambda_{\max}/10$ | $\mu = \lambda_{\max}/5$ | $\lambda_i$ | $\mu = \lambda_{\max}/50$ | $\mu = \lambda_{\max}/20$ |
| $1.79 \cdot 10^{-7}$ | $4.92 \cdot 10^{-10}$ | $3.40 \cdot 10^{-10}$ | $1.66 \cdot 10^{-8}$ | $8.87 \cdot 10^{-9}$ | $8.45 \cdot 10^{-9}$ |
| $1.80 \cdot 10^{-5}$ | $1.10 \cdot 10^{-11}$ | $1.85 \cdot 10^{-13}$ | $1.67 \cdot 10^{-6}$ | $2.74 \cdot 10^{-11}$ | $3.58 \cdot 10^{-11}$ |
| $6.89 \cdot 10^{-2}$ | $1.00 \cdot 10^{-15}$ | $2.61 \cdot 10^{-15}$ | $8.19 \cdot 10^{-3}$ | $2.07 \cdot 10^{-14}$ | $1.69 \cdot 10^{-14}$ |
| $1.65 \cdot 10^{-1}$ | $2.78 \cdot 10^{0}$ | $4.01 \cdot 10^{-5}$ | $1.64 \cdot 10^{-2}$ | $2.89 \cdot 10^{-4}$ | $2.36 \cdot 10^{-6}$ |
| $1.66 \cdot 10^{-1}$ | $3.33 \cdot 10^{0}$ | $1.48 \cdot 10^{-2}$ | $1.64 \cdot 10^{-2}$ | $9.55 \cdot 10^{-4}$ | $7.94 \cdot 10^{-9}$ |
| $1.67 \cdot 10^{-1}$ | $4.08 \cdot 10^{0}$ | $1.42 \cdot 10^{-2}$ | $2.09 \cdot 10^{-2}$ | $8.86 \cdot 10^{-7}$ | $1.08 \cdot 10^{-10}$ |
| $1.69 \cdot 10^{-1}$ | $8.77 \cdot 10^{0}$ | $8.77 \cdot 10^{-2}$ | $2.15 \cdot 10^{-2}$ | $3.08 \cdot 10^{-5}$ | $3.01 \cdot 10^{-9}$ |

products achieved is not that far from that achieved by the conjugate gradient alone. The important point to highlight is the substantial reduction of the dimension of the Krylov basis generated with this combination of CG with Chebyshev filters. This basis is indeed reduced by a factor of 3 in the case of IC($10^{-2}$), by a factor of about 6 with incomplete Cholesky no fill-in, and by a factor of about 10 with a simple Jacobi as the first level of preconditioner. This reduction is even stronger with smaller values for the filtering level $\varepsilon$. These different issues will be analyzed in more detail in terms of floating-point operations in section 5.

**3.2. Relevance of the Krylov basis.** In this section, we consider the case of the PDE1 problem. We evaluate the relevance of the information stored in the Krylov subspace $\mathbf{W}_k$ obtained from the CG method, after $k$ iterations of ChebFilterCG. To do so, we perform Ritz's spectral analysis of $\mathbf{W}_k$, and we also study the cosines of the principal angles between this Krylov subspace and the corresponding invariant subspace $\mathbf{U}_k$, associated with the $k$ smallest eigenvalues in the given matrix. $\mathbf{U}_k$ is computed with ARPACK [19].

**3.2.1. Ritz's spectral analysis.** The Ritz values $\delta_i$, $1 \leq i \leq k$, are the eigenvalues of the Rayleigh matrix $\mathbf{W}_k^T \mathbf{A} \mathbf{W}_k$, where $\mathbf{W}_k$ is a $n \times k$ matrix whose columns are the corresponding orthonormalized set of Krylov vectors. In Table 3.4, we give the relative residual corresponding to the smallest eigenvalues $\lambda_i$, $1 \leq i \leq 7$, which indicates the number of correct digits in each approximated eigenvalue $\delta_i$, $1 \leq i \leq 7$, for a level of filtering $\varepsilon = 10^{-4}$ and for different cutoff filtering values $\mu$. For instance, the conjugate gradient with a Chebyshev filter as preconditioner (ChebFilterCG) gives a good approximation of the three smallest eigenvalues (ten correct digits) for the preconditioned matrix with IC($10^{-2}$) and $\mu = \lambda_{\max}/10$. We can observe that, with $\mu = \lambda_{\max}/10$, the other four larger eigenvalues are not well approximated. This is simply because these eigenvalues fall directly in the interval $[\lambda_{\max}/10, \lambda_{\max}]$, where the Chebyshev polynomials used for preconditioning uniformly converge to 0.

This table proves the interest of the spectral information stored inside the Krylov basis generated by ChebFilterCG and which gives a good approximation of the spectral part in the range $]0, \mu[$.

**3.2.2. Cosines of the principal angles.** In Table 3.5, we give the cosines of the principal angles between the two subspaces $\mathbf{W}_k$ and $\mathbf{U}_k$, where $\mathbf{W}_k$ is the orthonormal basis for the Krylov subspace generated by the ChebFilterCG algorithm, and $\mathbf{U}_k$ is the subspace spanned by the $k$ smallest eigenvectors of the given matrix. The index $k$ refers to the number of iterations of the conjugate gradient with a Chebyshev filter as

Table 3.5

*Cosines of the principal angles between* Range $(\mathbf{W}_k)$ *and* Range $(\mathbf{U}_k)$*, the invariant subspace associated with the k smallest eigenvalues. The matrix is initially preconditioned with incomplete Cholesky with drop tolerance (*$10^{-2}$*), and the cutoff filtering value in the* ChebFilterCG *algorithm has been fixed to* $\mu = \lambda_{\max}/10$*. The problem considered is PDE1.*

| SVD($\mathbf{W}_k^T\,\mathbf{U}_k$) | | | |
|---|---|---|---|
| Values of filtering level $\varepsilon$ and corresponding size $k$ of the Krylov basis $\mathbf{W}_k$ | | | |
| $\varepsilon = 10^{-16}$ and $(k = 4)$ | $\varepsilon = 10^{-4}$ and $(k = 7)$ | $\varepsilon = 10^{-1}$ and $(k = 16)$ | |
| 1.000E+00 | 1.000E+00 | 1.000E+00 | 9.999E-01 |
| 1.000E+00 | 1.000E+00 | 1.000E+00 | 9.994E-01 |
| 9.999E-01 | 9.999E-01 | 1.000E+00 | 9.731E-01 |
| 9.538E-01 | 9.815E-01 | 9.999E-01 | 9.580E-01 |
| | 9.725E-01 | 9.999E-01 | 9.389E-01 |
| | 9.586E-01 | 9.993E-01 | 8.682E-01 |
| | 8.998E-01 | 9.999E-01 | 8.192E-01 |
| | | 9.999E-01 | 6.219E-01 |

the preconditioner. The computation is done using the singular value decomposition (SVD), as proposed in [15], since the singular values of $\mathbf{W}_k^T\,\mathbf{U}_k$ correspond to the cosines of the principal angles between the two subspaces Range $(\mathbf{W}_k)$ and Range $(\mathbf{U}_k)$. As we can see in Table 3.5, these principal angles stay very close to zero.

We deduce from these cosines of the principal angles that the two subspaces Range $(\mathbf{W}_k)$ and Range $(\mathbf{U}_k)$ are very collinear and especially for the directions which are related to the interval $]0, \mu[$.

**4. Reusing the filtered Krylov subspaces in further runs.** Let us now consider the case of a series of linear systems with different right-hand sides and the same coefficient matrix, viz.

$$\mathbf{A}\mathbf{x}_\ell = \mathbf{b}_\ell,\ \ \ell = 1, \ldots, s, \tag{4.1}$$

where $\mathbf{A}$ is a symmetric positive definite matrix in $\mathbb{R}^{n\times n}$; $\mathbf{x}_\ell$ and $\mathbf{b}_\ell$ are vectors of $\mathbb{R}^n$. The main idea is to solve the first system $\mathbf{A}\mathbf{x}_1 = \mathbf{b}_1$ in this sequence by means of the conjugate gradient with a Chebyshev filter as the preconditioner (ChebFilterCG) and to exploit the resulting complete Krylov basis to compute the solution of the remaining systems.

Once this Krylov basis $\mathbf{W}_k$ is obtained from the CG method, after $k$ iterations of ChebFilterCG when solving the first system in (4.1), we can use it in the Init-CG algorithm; see, e.g., [6, 7, 10, 12, 13, 23, 25, 26, 29] for the computation of the solutions in each of the following linear systems (4.1). We insist on the fact that we keep the same precomputed basis $\mathbf{W}_k$ for the solution of all of the remaining systems in the sequence. The Init-CG algorithm performs an oblique projection of the initial residual (i.e., a projection onto Span$(\mathbf{A}\mathbf{W}_k)$ along Ker$({\mathbf{W}_k}^T)$), in order to get the eigencomponents in the solution corresponding to the smallest eigenvalues, and then performs a classical conjugate gradient to compute the remaining part of the solution vector.

For the practical details, we use the conjugate gradient to solve $\mathbf{M}_1^{-1}\mathbf{A}\mathbf{x}_\ell = \mathbf{M}_1^{-1}\mathbf{b}_\ell$ $(\ell \geq 2)$ with the starting guess $\mathbf{x}_\ell^{(0)} = \mathbf{W}_k(\mathbf{W}_k^T\mathbf{A}\mathbf{W}_k)^{-1}\mathbf{W}_k^T\mathbf{b}_\ell$, where $\mathbf{W}_k$, of dimension $k$, is equal to the Krylov basis obtained after $k$ iterations of the ChebFilterCG when solving the first system with the preconditioned matrix $\mathbf{M}_1^{-1}\mathbf{A}$. Our Krylov basis $\mathbf{W}_k$ generated by ChebFilterCG is formed by the search direction vectors and not by the residuals. In this case $\mathbf{W}_k$ is already an $\mathbf{A}$-orthogonal basis

Algorithm 4.1:

```
Begin
    1. $[\mathbf{x}_1, \mathbf{W}] = \textsc{ChebFilterCG}\ (\mathbf{A}, \mathbf{b}_1, \mathbf{x}^{(0)}, \mathrm{tol}_1, \mathbf{M}_1)$
    2. $\mathbf{A}_c = \mathbf{W}^T\mathbf{A}\mathbf{W}$ is a diagonal matrix
    3. For $\ell = 2, \ldots, s$ Do
        i. $\mathbf{x}^{(0)} = \mathbf{W}\mathbf{A}_c^{-1}\mathbf{W}^T\mathbf{b}_\ell$
        ii. $\mathbf{x}_\ell = \mathrm{PCG}\ (\mathbf{A}, \mathbf{b}_\ell, \mathbf{x}^{(0)}, \mathrm{tol}_2, \mathbf{M}_1)$
    4. EndFor
End
```

of the Krylov subspace, and $\mathbf{A}_c = \mathbf{W}_k^T\mathbf{A}\mathbf{W}_k$ is a diagonal matrix. Note that we do not project the given matrix in any way but that we use the preconditioned conjugate gradient with the original matrix and the first level of preconditioning $\mathbf{M}_1$ and with an initial guess which, we expect, will remove all of the difficulties that the conjugate gradient algorithm would encounter otherwise, e.g., the plateaus that can be observed in the convergence history. This approach is resumed in Algorithm 4.1.

To illustrate this strategy, we consider the case of the PDE1 problem, and we present some numerical experiments where we solve a first system and then a second system. The initial guess is $\mathbf{x}^{(0)} = 0$, and the first right-hand side $\mathbf{b}_1$ is chosen so that the solution $\mathbf{x}^\star$ of the linear system is the vector of all ones and the solution in ChebFilterCG of this first linear system is stopped when the $\mathbf{A}$-norm of the error is below $10^{-9}$. The second right-hand side $\mathbf{b}_2$ is chosen so that the solution of the linear system is a vector with normally distributed random numbers. In these runs, we have monitored the $\mathbf{A}$-norm of the error down to machine precision in order to illustrate and study the complete convergence history. For all of the numerical experiments reported in this work, the ChebFilterCG algorithm is used for the first right-hand side $\mathbf{b}_1$, while the Init-CG algorithm is used for the second right-hand side $\mathbf{b}_2$.

Figures 4.1 and 4.2 summarize the improvement when reusing the Krylov subspaces generated by the ChebFilterCG algorithm in the solution of linear systems with the same matrix and with changing right-hand sides. We plot the convergence history, in the Init-CG method, of the error measured in the energy norm, respectively, for the Jacobi preconditioner and for the incomplete Cholesky with no fill-in preconditioner. We also plot the curve of the errors measured in the energy norm when the conjugate gradient method is used without this deflation of the initial residual.

Figure 4.1 shows the effectiveness of the Krylov basis generated by the conjugate gradient with a Chebyshev filter as the preconditioner for different values of filtering level $\varepsilon$. We observe that, for a fixed value of $\mu$, the convergence history of Init-CG for different values of filtering level $\varepsilon$ yields a constant numerical behavior, since all of these curves coincide almost completely, except perhaps after stagnation. We can also observe that the number of iterations is reduced by a factor of about 4 to reach an $\mathbf{A}$-norm of the error of $10^{-8}$, which illustrates the relevance of the information contained in the Krylov subspaces generated by the conjugate gradient with a Chebyshev as the preconditioner. An important property of the ChebFilter preconditioner is that it does not need a level of filtering $\varepsilon$ close to $10^{-16}$ for the Krylov basis to be efficient.

In Figure 4.2, we consider the case when the filtering value is fixed (for instance, $\varepsilon = 10^{-1}$) and the cutoff value $\mu$ is varied. The convergence history of the Init-

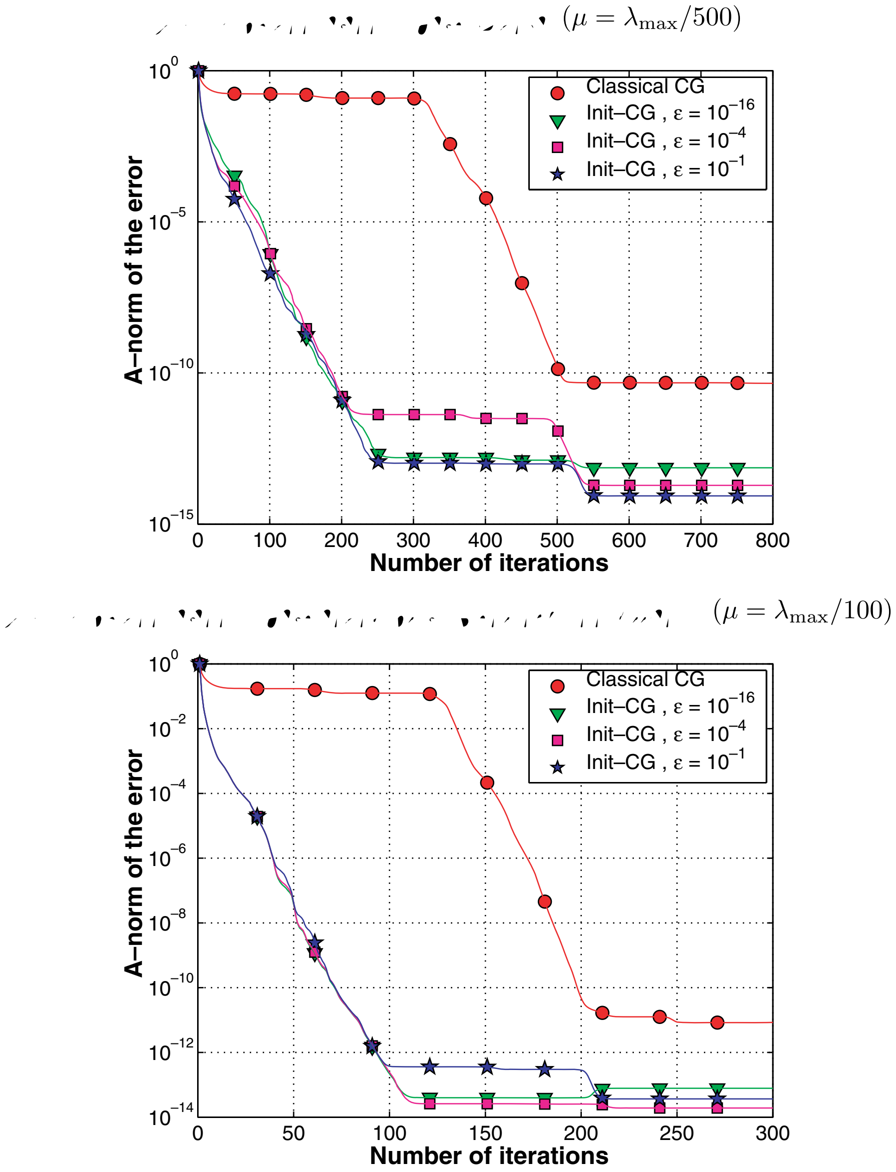


FIG. 4.1. *Convergence history of the* INIT-CG *algorithm when solving the second linear system* $\mathbf{A}\mathbf{x}_2 = \mathbf{b}_2$ *and exploiting the Krylov basis generated by the conjugate gradient with a Chebyshev filter as the preconditioner from the first solution. This for different values of* $\varepsilon$ *and for a fixed value of* $\mu$*. The problem considered is PDE*1.

CG with the Krylov basis generated by the conjugate gradient with a Chebyshev as the preconditioner (CHEBFILTERCG), for different values of $\mu$, exhibits the same numerical behavior. Although these runs are restricted to only two systems, the gains obtained can easily be extended to longer sequences of linear systems with the same coefficient matrix.

Finally, the different histograms show that the level of filtering $\varepsilon$ and the cutoff eigenvalue $\mu$ do not have a big effect on the quality of the Krylov basis, since the

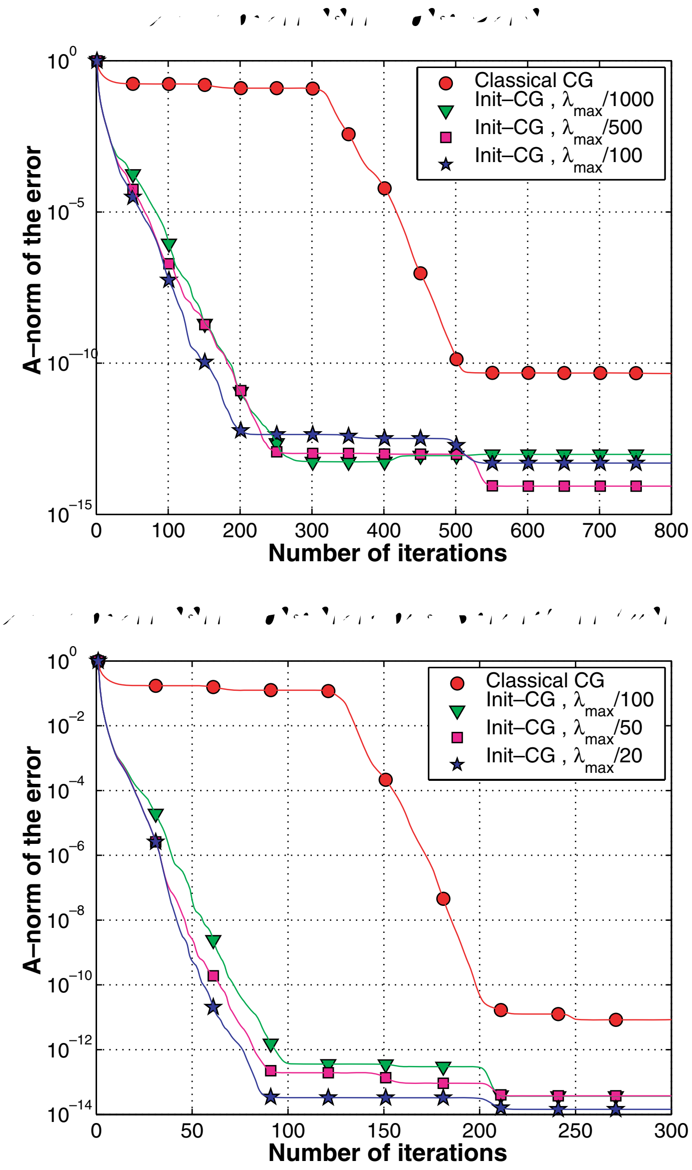


FIG. 4.2. *Convergence history of* INIT-CG *when solving the second linear system* $\mathbf{A}\mathbf{x}_2 = \mathbf{b}_2$ *and exploiting the Krylov basis generated by the conjugate gradient with a Chebyshev as the preconditioner from the first solution. This for* $\varepsilon = 10^{-1}$ *and different values of* $\mu$. *The problem considered is PDE1.*

convergence behavior of the INIT-CG algorithm does not vary much for a basis **W** obtained with different values for these parameters. Nevertheless, these parameters certainly act on the size of the resulting basis **W**, which remains at any rate relatively small and very rich with respect to the smallest eigenvalues and associated eigenvectors.

**5. Practical considerations.** We consider the operations count for the CHEB-FILTERCG and INIT-CG algorithms. All costs are evaluated in number of floating-point operations. The cost for a matrix-vector product is given by

$$\mathcal{C}_{\mathbf{A}} \approx 2\, \mathit{nnz}\,(\mathbf{A}) - n, \tag{5.1}$$

where $\mathit{nnz}\,(\mathbf{A})$ is the number of nonzero elements in $\mathbf{A}$.

In the conjugate gradient method with a Chebyshev filter as the preconditioner, we use the Chebyshev filtering polynomials in the matrix $\mathbf{A}$ at each conjugate gradient iteration to reduce, under the level of filtering $\varepsilon$, the eigencomponents in the residual associated with all eigenvalues in the range $[\mu, \lambda_{\max}]$ relatively to the others. The cost of one Chebyshev filtering step (one iteration in Algorithm 2.1) involves one sparse matrix-vector product with $\mathbf{A}$ and three vector updates (`_AXPY`). This cost can be estimated by

$$\mathcal{C}_{\mathtt{ChebFilter}} = \mathcal{C}_{\mathbf{A}} + 3\,\mathcal{C}_{\mathtt{\_AXPY}}, \text{ where } \mathcal{C}_{\mathtt{\_AXPY}} \approx 2\,n. \tag{5.2}$$

In addition to the sparse matrix-vector product with $\mathbf{A}$, the conjugate gradient method merely adds two dot products (`_DOT`) and three vector updates (`_AXPY`). The cost of these operations is given by

$$\mathcal{C}_{\mathtt{CG}} = \mathcal{C}_{\mathbf{A}} + 2\,\mathcal{C}_{\mathtt{\_DOT}} + 3\,\mathcal{C}_{\mathtt{\_AXPY}}, \text{ where } \mathcal{C}_{\mathtt{\_DOT}} \approx 2\,n. \tag{5.3}$$

Thus, the total cost of our scheme is of order

$$\mathcal{C}_{\mathtt{ChebFilterCG}} = \Big(\mathcal{C}_{\mathtt{CG}} + \mathtt{ChebIt} \times \mathcal{C}_{\mathtt{ChebFilter}}\Big) \times \mathtt{CGIt}, \tag{5.4}$$

where `ChebIt` is the number of Chebyshev filtering steps performed at each conjugate gradient iteration, and `CGIt` is the number of conjugate gradient iterations.

**5.1. Cost benefits and remarks.** In this paragraph, we show the cost benefits of our technique for solving large linear systems with different right-hand sides and the same coefficient matrix (4.1). We solve the first system by the conjugate gradient with a Chebyshev filter as the preconditioner, and we use the resulting complete Krylov basis $\mathbf{W}$ in the INIT-CG algorithm (see section 4) for the computation of the solutions in each of the following global iterations.

As mentioned at the beginning of section 3, a first level left preconditioner $\mathbf{M}_1$ is also used, whose purpose is to better cluster the spectrum of our iteration matrix. The cost of the multiplication of $\mathbf{M}_1$ by a vector is represented by $\mathcal{C}_{\mathbf{M}_1}$. Since $\mathbf{M}_1$ is constructed in our experiments by means of the incomplete Cholesky factorization or Jacobi, $\mathcal{C}_{\mathbf{M}_1}$ can be estimated as

$$\mathcal{C}_{\mathbf{M}_1} \approx 4\, \mathit{nnz}\,(\mathbf{L}) - 2\,n, \tag{5.5}$$

where $\mathit{nnz}\,(\mathbf{L})$ is the number of nonzero elements in the factor $\mathbf{L}$ from the incomplete Cholesky factorization or Jacobi, and $n$ is the size of the linear system. The cost of this first level of preconditioning $\mathbf{M}_1$ must be incorporated in the floating-point counts detailed in section 5. This simply results in adding $\mathcal{C}_{\mathbf{M}_1}$ explicitly at each Chebyshev filtering step and each conjugate gradient (CG) and (INIT-CG) iteration (see (5.2), (5.3), and (5.8)).

Our implementation of the INIT-CG algorithm performs, at the beginning of the conjugate gradient algorithm, an oblique projection of the right-hand side $\mathbf{b}$ to construct the starting guess $\mathbf{x}^{(0)} = \mathbf{W}_k \mathbf{A}_c^{-1} \mathbf{W}_k^T \mathbf{b}$, where $\mathbf{A}_c = \mathbf{W}_k^T \mathbf{A} \mathbf{W}_k$ is a matrix of order $k$, and sets $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$ and $\mathbf{p}^{(0)} = \mathbf{r}^{(0)}$. The computation of $\mathbf{A}_c$ is a BLAS3 operation whose cost is given by

$$\mathcal{C}_{\mathbf{A}_c} \approx n\,k^2 + 2\,\mathit{nnz}\,(\mathbf{A})\,k - n\,k. \tag{5.6}$$

We note that $\mathbf{A}_c$ is factored once at the beginning of the Init-CG algorithm, and these factors are used for any subsequent solution with $\mathbf{A}_c$.

We mention again that, if the basis $\mathbf{W}_k$ is formed from the set of $\mathbf{A}$-orthogonal Krylov vectors in the CG iterations, $\mathbf{A}_c$ is diagonal and does not need to be factored, and this cost can be ignored in this case. We have incorporated this cost to be general and to consider also the case when $\mathbf{W}_k$ is obtained with other techniques. In particular, for a basis coming from Lanczos, $\mathbf{A}_c$ is tridiagonal. If $\mathbf{W}_k$ is obtained with any of the classical spectral factorization techniques, $\mathbf{A}_c$ is also diagonal as with the $\mathbf{A}$-orthogonal vectors in CG. However, if $\mathbf{W}_k$ is just a very rough approximation to all of these, $\mathbf{A}_c$ should be considered dense.

With $\mathbf{A}_c$ precomputed and factored, the computation of an oblique projection and an update of the initial residual can be done using common BLAS2 operations. The total cost of these operations is represented by

$$\mathcal{C}_{\mathrm{Proj}} \approx 4\,(k+1)\,n, \tag{5.7}$$

where the costs in $\mathcal{O}(k^3)$ operations have been neglected.

In addition, each iteration also requires one sparse matrix-vector product with $\mathbf{A}$, two dot products (`_DOT`), and three vector updates (`_AXPY`) (see (5.1) and (5.3)). The total cost of the Init-CG algorithm is represented by

$$\mathcal{C}_{\texttt{InitCG}} = \Big( \underbrace{\mathcal{C}_{\mathbf{A}} + \mathcal{C}_{\mathrm{Proj}}}_{\text{At the beginning}} \Big) + \Big( \underbrace{\mathcal{C}_{\mathbf{A}} + 3\,\mathcal{C}_{\texttt{\_AXPY}} + 2\,\mathcal{C}_{\texttt{\_DOT}}}_{\text{At each iteration}} \Big) \times \texttt{Nit}, \tag{5.8}$$

where `Nit` is the total number of Init-CG iterations.

In Table 5.1, we report the number of floating-point operations ($\mathcal{C}_{\texttt{ChebFilterCG}}$) of the ChebFilterCG algorithm (in millions, `Mflop`), when solving the first linear system $\mathbf{A}\mathbf{x}_1 = \mathbf{b}_1$ in the sequence (4.1). We also illustrate the cost benefit of the Init-CG algorithm when solving the second linear system $\mathbf{A}\mathbf{x}_2 = \mathbf{b}_2$ and exploiting the spectral information obtained from the first solution. We stop the conjugate gradient iterations when $\|\mathbf{x}^\star - \mathbf{x}^{(k)}\|_{\mathbf{A}} \le 10^{-9}\ \|\mathbf{x}^\star\|_{\mathbf{A}}$. For each level of filtering $\varepsilon$, we indicate in this table the cost $\mathcal{C}_{\mathbf{A}_c}$ (see formula (5.6)) in `Mflop`, the size $k$ of the resulting Krylov basis from the solution of the first linear system with ChebFilterCG (this number $k$ also corresponds to `CGIt` in Table 3.3), the number of Init-CG iterations (`Nit`), and the number of floating-point operations ($\mathcal{C}_{\texttt{InitCG}}$) to reach the above stopping criterion for the $\mathbf{A}$-norm of the error.

We also compute the number of right-hand sides that have to be considered in subsequent consecutive solutions before the extra cost $\mathcal{C}_{\texttt{ChebFilterCG}}$, for solving the first linear system and computing the Krylov basis $\mathbf{W}$, is compensated. This is indicated as the number of amortization vectors (`Amor`). To compute this number, we must compare the number of floating-point operations needed to reach the given level of the $\mathbf{A}$-norm of the error with the conjugate gradient algorithm and the number of floating-point operations to reach the same level with the Init-CG algorithm. Note that $\mathbf{A}_c$ is factored once at the end of the first solution, and these factors are used for any solution with $\mathbf{A}_c$. Consequently, the number of amortization vectors (`Amor`)

TABLE 5.1

*Number of floating-point operations ($\mathcal{C}_{\texttt{ChebFilterCG}}$) of the* CHEBFILTERCG *algorithm and cost and benefits of* INIT-CG *for different values of the level of filtering $\varepsilon$ and for different cutoff filtering values. The iterations of the* INIT-CG *algorithm (*`Nit`*) are stopped when the* **A***-norm of the error is below* $10^{-9}$*. The problem considered is PDE*1.

PDE1 preconditioned with Jacobi

CG without Chebyshev preconditioning performs 485 iterations

with a cost of 95.79 `Mflop`

| $\varepsilon$ | $\mu = \lambda_{\max}/500$ | | | | | | $\mu = \lambda_{\max}/100$ | | | | | | $\mu = \lambda_{\max}/50$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` |
| $10^{-16}$ | 349.68 | 0.71 | 5 | 152 | 30.19 | 4 | 345.56 | 2.09 | 11 | 160 | 31.96 | 4 | 353.85 | 3.68 | 16 | 153 | 30.74 | 5 |
| $10^{-8}$ | 214.39 | 0.90 | 6 | 152 | 30.22 | 2 | 258.21 | 3.68 | 16 | 142 | 28.56 | 3 | 264.25 | 6.57 | 23 | 131 | 26.61 | 3 |
| $10^{-4}$ | 149.03 | 1.33 | 8 | 163 | 32.46 | 1 | 187.00 | 6.11 | 22 | 130 | 26.38 | 2 | 186.29 | 10.89 | 31 | 140 | 28.58 | 2 |
| $10^{-2}$ | 121.92 | 2.38 | 12 | 159 | 31.79 | 1 | 140.43 | 10.24 | 30 | 127 | 26.04 | 1 | 144.17 | 19.13 | 43 | 126 | 26.26 | 1 |
| $10^{-1}$ | 111.03 | 4.82 | 19 | 159 | 32.02 | 1 | 112.92 | 18.35 | 42 | 126 | 26.23 | 1 | 119.44 | 33.78 | 59 | 126 | 26.76 | 1 |

PDE1 preconditioned with incomplete Cholesky (no fill-in)

CG without Chebyshev preconditioning performs 191 iterations

with a cost of 55.52 `Mflop`

| $\varepsilon$ | $\mu = \lambda_{\max}/100$ | | | | | | $\mu = \lambda_{\max}/50$ | | | | | | $\mu = \lambda_{\max}/20$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` |
| $10^{-16}$ | 196.97 | 0.54 | 4 | 62 | 17.31 | 4 | 207.97 | 0.90 | 6 | 60 | 17.38 | 4 | 197.13 | 1.57 | 9 | 60 | 17.78 | 4 |
| $10^{-8}$ | 126.44 | 0.71 | 5 | 62 | 17.34 | 2 | 143.98 | 1.33 | 8 | 60 | 17.44 | 2 | 149.36 | 2.68 | 13 | 60 | 17.90 | 2 |
| $10^{-4}$ | 93.17 | 1.11 | 7 | 62 | 17.41 | 2 | 103.46 | 2.10 | 11 | 58 | 17.53 | 2 | 108.37 | 4.42 | 18 | 59 | 17.78 | 2 |
| $10^{-2}$ | 80.54 | 2.10 | 11 | 62 | 17.53 | 1 | 78.58 | 3.33 | 15 | 56 | 16.79 | 1 | 85.42 | 7.54 | 25 | 48 | 14.77 | 1 |
| $10^{-1}$ | 75.56 | 4.42 | 18 | 62 | 17.77 | 1 | 69.44 | 6.11 | 22 | 52 | 15.84 | 1 | 74.02 | 13.34 | 35 | 48 | 15.01 | 1 |

PDE1 preconditioned with incomplete Cholesky (tol=$10^{-2}$)

CG without Chebyshev preconditioning performs 54 iterations

with a cost of 22.91 `Mflop`

| $\varepsilon$ | $\mu = \lambda_{\max}/20$ | | | | | | $\mu = \lambda_{\max}/10$ | | | | | | $\mu = \lambda_{\max}/5$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` | $\mathcal{C}_{\texttt{ChebFilterCG}}$ `Mflop` | $\mathcal{C}_{\mathbf{A}_c}$ `Mflop` | `k` | `Nit` | $\mathcal{C}_{\texttt{InitCG}}$ `Mflop` | `Amor` |
| $10^{-16}$ | 131.99 | 0.54 | 4 | 21 | 8.99 | 8 | 92.74 | 0.54 | 4 | 18 | 7.71 | 5 | 78.65 | 0.71 | 5 | 18 | 7.74 | 4 |
| $10^{-8}$ | 69.20 | 0.54 | 4 | 21 | 8.99 | 4 | 60.99 | 0.71 | 5 | 18 | 7.74 | 4 | 57.91 | 1.11 | 7 | 18 | 7.81 | 3 |
| $10^{-4}$ | 54.35 | 0.90 | 6 | 21 | 9.05 | 3 | 46.92 | 1.11 | 7 | 18 | 7.81 | 3 | 42.67 | 1.57 | 9 | 18 | 7.87 | 2 |
| $10^{-2}$ | 51.34 | 1.81 | 10 | 21 | 9.19 | 3 | 39.56 | 1.81 | 10 | 18 | 7.92 | 2 | 36.13 | 2.68 | 13 | 18 | 8.00 | 2 |
| $10^{-1}$ | 47.57 | 3.32 | 15 | 21 | 9.35 | 3 | 38.18 | 3.68 | 16 | 18 | 8.10 | 2 | 33.89 | 4.04 | 17 | 18 | 8.13 | 2 |

is given by

$$\texttt{Amor} = \left\lceil \frac{\mathcal{C}_{\texttt{ChebFilterCG}} + \mathcal{C}_{\mathbf{A}_c} - \mathcal{C}_{\texttt{CG}}}{\mathcal{C}_{\texttt{CG}} - \mathcal{C}_{\texttt{InitCG}}} \right\rceil. \tag{5.9}$$

This formula takes into account the fact that construction of $\mathbf{W}_k$ also provides the solution for the first linear system $\mathbf{A}\mathbf{x}_1 = \mathbf{b}_1$ in the sequence (4.1).

For instance, to reach an **A**-norm of the error of $10^{-9}$, the conjugate gradient algorithm performs 485 iterations in the case of the matrix preconditioned with Jacobi, with a cost of 95.79 `Mflop`, and 191 iterations in the case of the matrix preconditioned with incomplete Cholesky (no fill-in), with a cost of 55.52 `Mflop`, and 54 iterations in the case of the matrix preconditioned with incomplete Cholesky with drop tolerance $10^{-2}$, with a cost of 22.91 `Mflop`.

If we consider the case of the matrix preconditioned with incomplete Cholesky with drop tolerance $10^{-2}$, with a level of filtering $\varepsilon = 10^{-4}$ and a parameter $\mu = \lambda_{\max}/10$, in particular, 46.92 `Mflop` are needed for solving the first linear system and computing the Krylov basis $\mathbf{W}_k$, out of which INIT-CG convergence is achieved in 18 iterations (see Table 5.1), i.e., a reduction of 66% compared to the run of the conjugate gradient algorithm. The 46.92 extra `Mflop` are paid back after three consecutive runs of INIT-CG (see Table 5.1). In this particular case, the eigenvalues of the preconditioned matrix are indeed well clustered (see Tables 3.2 and 3.4) and enable a strong reduction in the number of iterations with a subspace $\mathbf{W}_k$ of dimension 4 already. This shows how this spectral approximation technique can be an effective complementary tool to other classical preconditioning techniques.

Table 5.1 also shows the impact of varying the filtering level $\varepsilon$ and the parameter $\mu$ on the actual number of amortization vectors. We can observe that the number of amortization vectors (`Amor`) does not change dramatically for the INIT-CG algorithm for a given level of filtering $\varepsilon$ and that in all cases the amortization of the cost of the computation of the Krylov basis $\mathbf{W}_k$ is very fast.

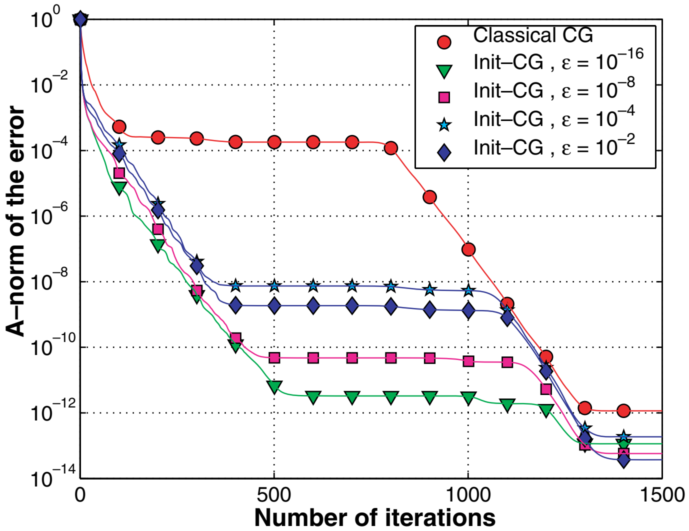
In summary, when the sequence of several linear systems with the same matrix but different right-hand sides is very long, the strategy of choice is to compute the solution of the first linear system by the conjugate gradient with a Chebyshev filter as the preconditioner using a level of filtering $\varepsilon$ close $10^{-16}$, in order to minimize the size of the Krylov basis $\mathbf{W}_k$, and to exploit the resulting complete Krylov basis in the INIT-CG algorithm to compute the solution of remaining systems. Indeed, a level of filtering very close to machine precision enables us to obtain, on the one hand, a smaller dimensional Krylov basis and at the same time less memory requirements and, on the other hand, an optimal cost of the INIT-CG algorithm or not far from the optimal one.

### 5.2. Application to the anisotropic problem PDE2 of larger size.

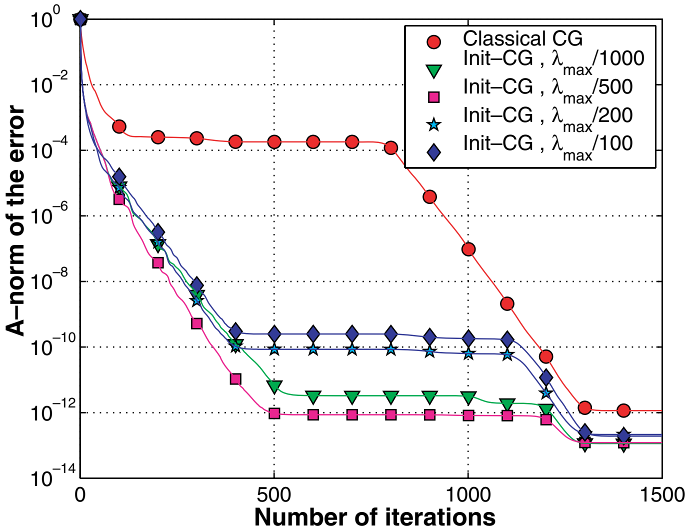
We have also considered the larger test problem PDE2 introduced in section 3. In this case, we have used the incomplete Cholesky decomposition of $\mathbf{A}$ with no fill-in as the first level of the left preconditioner $\mathbf{M}_1 = \mathbf{L}\mathbf{L}^T$, where $\mathit{nnz}(\mathbf{L}) = 643{,}605$.

In Figure 5.1, we plot the convergence history of the error measured in the energy norm in the INIT-CG algorithm (when solving $\mathbf{A}\mathbf{x}_2 = \mathbf{b}_2$), with different values of the level of filtering $\varepsilon$ and a fixed value of $\mu$ on the top figure and with different values of the cutoff value $\mu$ and a fixed filtering level $\varepsilon$ on the bottom figure. We also plot the history of the errors measured in the energy norm when the conjugate gradient method is used without the deflation of the initial residual. We can observe that the number of iterations is reduced by a factor of 4 to reach an $\mathbf{A}$-norm of the error of $10^{-8}$, which shows the effectiveness of the information stored inside the orthonormal basis of the filtered Krylov subspace.

In this larger size test case, we have stopped all iterations at a level of $10^{-8}$. It can be observed in Figure 5.1 that the phenomenon of plateaus occurs again at that level of convergence of the INIT-CG algorithm. This is due to the sensitivity of the oblique projector to the total size of the linear system [2]. If better accuracy is required, however, a possibility is to use the TWO-GRID cycle scheme presented in [5, 12, 29] or to perform iterative refinement in the usual way. In [12], [29, section 3.3.4], for instance, the effectiveness of this strategy has been experimented with either



(a) [illegible] $\mu = \lambda_{\max}/1000$



(b) [illegible] $\varepsilon = 10^{-16}$

FIG. 5.1. *Convergence history of the preconditioned conjugate gradient without the deflation of the initial residual and the* INIT-CG *algorithm with different values of the level of filtering $\varepsilon$ and for different values of $\mu$, on the PDE2 problem.*

conjugate gradient or Chebyshev polynomials or even a Richardson iteration as the smoother.

In Table 5.2, we display the number of Chebyshev filtering steps (`ChebIt`) performed at each conjugate gradient step, the number of conjugate gradient iterations (`CGIt`), and the number of floating-point operations ($\mathcal{C}_{\texttt{ChebFilterCG}}$) (in thousand million, `Gflop` ), and this with different values of level of filtering $\varepsilon$. The cost of the first level of preconditioning $\mathbf{M}_1$ has also been incorporated in the floating-point counts of the conjugate gradient with a Chebyshev filter as the preconditioner.

Table 5.2

*Cost of* ChebFilterCG *for different values of the level of filtering $\varepsilon$ and for different cutoff filtering values. The* ChebFilterCG *iterations are stopped when the* **A***-norm of the error is below* $10^{-8}$*. The problem considered is PDE2.*

| | CG without Chebyshev preconditioning performs 1060 iterations with a cost of 6.31 Gflop | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\mu = \lambda_{\max}/1000$ | | | $\mu = \lambda_{\max}/500$ | | | $\mu = \lambda_{\max}/200$ | | |
| Value of $\varepsilon$ | ChebIt | CGIt | $\mathcal{C}_{\texttt{ChebFilterCG}}$ Gflop | ChebIt | CGIt | $\mathcal{C}_{\texttt{ChebFilterCG}}$ Gflop | ChebIt | CGIt | $\mathcal{C}_{\texttt{ChebFilterCG}}$ Gflop |
| $10^{-16}$ | 594 | 8 | 25.28 | 420 | 11 | 24.60 | 265 | 18 | 25.44 |
| $10^{-8}$ | 303 | 11 | 17.76 | 214 | 16 | 18.28 | 135 | 25 | 18.07 |
| $10^{-4}$ | 157 | 15 | 12.59 | 111 | 22 | 13.10 | 70 | 36 | 13.60 |
| $10^{-2}$ | 84 | 21 | 9.49 | 60 | 31 | 10.06 | 38 | 49 | 10.18 |

Table 5.3

*Cost and benefits of* Init-CG *for different values of the level of filtering $\varepsilon$ and for different cutoff filtering values. The iterations of the* Init-CG *algorithm are stopped when the* **A***-norm of the error is below* $10^{-8}$*. The problem considered is PDE2.*

| | CG without Chebyshev preconditioning performs 1060 iterations with a cost of 6.31 Gflop | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mu = \lambda_{\max}/1000$ | | | | | $\mu = \lambda_{\max}/500$ | | | | | $\mu = \lambda_{\max}/200$ | | | | |
| Value of $\varepsilon$ | $\mathcal{C}_{\mathbf{A}_c}$ Gflop | k | Nit | $\mathcal{C}_{\texttt{InitCG}}$ Gflop | Amor | $\mathcal{C}_{\mathbf{A}_c}$ Gflop | k | Nit | $\mathcal{C}_{\texttt{InitCG}}$ Gflop | Amor | $\mathcal{C}_{\mathbf{A}_c}$ Gflop | k | Nit | $\mathcal{C}_{\texttt{InitCG}}$ Gflop | Amor |
| $10^{-16}$ | 0.02 | 8 | 274 | 1.64 | 5 | 0.04 | 11 | 230 | 1.38 | 4 | 0.09 | 18 | 265 | 1.59 | 5 |
| $10^{-8}$ | 0.04 | 11 | 291 | 1.74 | 3 | 0.07 | 16 | 246 | 1.48 | 3 | 0.15 | 25 | 252 | 1.55 | 3 |
| $10^{-4}$ | 0.06 | 15 | 343 | 2.06 | 2 | 0.12 | 22 | 259 | 1.56 | 2 | 0.28 | 36 | 312 | 1.88 | 2 |
| $10^{-2}$ | 0.11 | 21 | 320 | 1.92 | 1 | 0.21 | 31 | 324 | 1.96 | 1 | 0.48 | 49 | 341 | 2.07 | 1 |

In Table 5.3, we illustrate the cost benefit of the Init-CG algorithm. We stop the conjugate gradient iterations when the **A**-norm of the error is below $10^{-8}$. The preconditioned conjugate gradient method (no initial deflation) performs 1060 iterations with a cost of 6.31 `Gflop`. We also indicate the cost $\mathcal{C}_{\mathbf{A}_c}$ in `Gflop`, the number of conjugate gradient iterations (`Nit`), the number of floating-point operations ($\mathcal{C}_{\texttt{InitCG}}$) in `Gflop`, and the number of amortization vectors (`Amor`). All of this information is given for each level of filtering $\varepsilon$. We derive, from these data, the same general conclusions as with the smaller size test cases in the previous section. In particular, for a fixed value of the level of filtering $\varepsilon$, the number of floating-point operations (`Gflop`) does not change dramatically when varying the parameter $\mu$. In Table 5.3, we can observe that the number of iterations is cut from 1060 to under 349 and as low as 230.

**5.3. Comments about the parameter $\boldsymbol{\lambda}_{\mathbf{max}}$ and its estimation.** In the various test examples used for discussion in this article, the value of $\lambda_{\max}(\mathbf{A})$ has been obtained rather accurately with the use of ARPACK [19]. The purpose of this spectral precomputation of the spectrum of **A** was to ease the analysis of the various results and to understand better the behavior of our algorithm in the different situations.

Of course, this has a cost, and it is not mandatory that either parts of the spectrum of **A** or even $\lambda_{\max}(\mathbf{A})$ only be accurately precomputed. An estimate $\widetilde{\lambda}_{\max}$ of $\lambda_{\max}$ should be sufficient as a parameter in this approach. The only requirement, to keep

Table 5.4
$\lambda_{\max}$ *and its estimation* $\widetilde{\lambda}_{\max}$ *for the PDE*1 *problem.*

| First level preconditioner | $\lambda_{\max}$ | $\widetilde{\lambda}_{\max}$ | $2 \times \widetilde{\lambda}_{\max}$ |
|---|---|---|---|
| $\mathbf{M}_1 = \mathbf{I}$ | $9.6 \cdot 10^6$ | $5.18 \cdot 10^6$ | $1.03 \cdot 10^7$ |
| $\mathbf{M}_1 = \text{diag}(\mathbf{A})$ | 2.08 | 1.28 | 2.56 |
| $\mathbf{M}_1 = \text{IC}(0)$ | 1.6 | 1.01 | 2.03 |
| $\mathbf{M}_1 = \text{IC}(10^{-2})$ | 1.1 | $9.78 \cdot 10^{-1}$ | 1.95 |

the Chebyshev polynomial filters bounded on the spectrum of **A**, is that this estimate $\widetilde{\lambda}_{\max}$ be an upper bound of the actual largest eigenvalue. A cheap way to achieve this is to perform a few steps of the power method and to multiply the final Rayleigh quotient by some factor to ensure the overestimation. Let's suppose, for instance, that we multiply by "2" a value giving the order of magnitude of $\lambda_{\max}$. The consequence of this will be that the choice of the cutoff value $\tilde{\mu} = \widetilde{\lambda}_{\max}/\gamma$ ($\gamma = 10$ or 100, for instance) will be larger than the actual value $\lambda_{\max}/\gamma$. Therefore, the effective filtering interval will become $[\widetilde{\mu}, \lambda_{\max}]$ (and not $[\mu, \lambda_{\max}]$ as in our experiments), because the behavior of Chebyshev on $]\lambda_{\max}, \widetilde{\lambda}_{\max}]$ does not have any impact on the preconditioning of **A**.

Still, and this is illustrated in the results in Table 3.3, choosing $\lambda_{\max}/100$ or $\lambda_{\max}/50$ as in the case of the matrix preconditioned with incomplete Cholesky (no fill-in) does not change much the size of the resulting Krylov basis in the ChebFilterCG algorithm. The major impact is the increase of the total number of Chebyshev steps for the same resulting Krylov basis, and the amortization will take a little longer, but not a lot more, as shown in Table 5.1. We can conclude that an overestimation $\widetilde{\lambda}_{\max}$ of $\lambda_{\max}$ is not very crucial with respect to the conclusions and observations made in the previous sections.

A more crucial issue is to find the range of values of $\mu$ in which the behavior of the algorithm is rather stable, as observed in the various experiments. This is specifically linked to the actual eigenvalue distribution of the given iteration matrix and depends on the problem and on the first level of preconditioner (when used). We must not forget also that the total number of systems to solve in the sequence may lead to different strategies when choosing $\mu$ or $\varepsilon$ to minimize the total amount of work.

In the previous discussions, we have not included the cost of the precomputation of $\widetilde{\lambda}_{\max}$ for two reasons. The first one is that we have not investigated the most powerful techniques for such computation, such as, for instance, power method, polynomial, and Krylov techniques and combinations of these. The second reason is that, in some cases, as with row projection techniques [1, 4, 9], for instance, that also yield a symmetrizable iteration matrix, the preconditioning technique can readily provide a more or less sharp estimate of $\lambda_{\max}(\mathbf{A})$. Finally, at the expense of one usual CG run, we can be sure to get a "[illegible]" estimate of $\lambda_{\max}$. Therefore, we can consider that the cost for precomputing $\lambda_{\max}$ will at most increase the amortization value of "1."

As an illustration, we indicate in Table 5.4 the values of $\lambda_{\max}$ and the corresponding estimation $\widetilde{\lambda}_{\max}$ obtained in the case of the PDE1 problem for each first level preconditioner. The largest eigenvalue $\lambda_{\max}$ is computed with ARPACK [19] and $\widetilde{\lambda}_{\max}$ results from three iterations of the power method with a random initial vector. We also report in this table the "[illegible]" estimation of $\lambda_{\max}$, which corresponds to $2 \times \widetilde{\lambda}_{\max}$ and which is not too far from the largest eigenvalue.

**6. Conclusion.** In this paper, we have proposed a solution technique suited for the solution of a sequence of linear systems with the same matrix but changing right-hand sides. This technique uses Chebyshev filtering polynomials, applied only to a part of the spectrum of the coefficient matrix, as a preconditioner that helps the conjugate gradient method to generate a low dimensional Krylov basis that is very rich with respect to the smallest eigenvalues and associated eigenvectors. The proposed approach is guaranteed to put a large number of eigenvalues near one, without degrading the distribution of the smallest ones.

We have illustrated, on a set of MATLAB examples, the behavior of this technique on sparse linear systems arising from the discretization via a finite element of some 2D heterogeneous and anisotropic diffusion PDE problems. An analysis of the computational costs in section 5.1 has shown that the gains can be rather substantial and the cost for precomputing the Krylov basis can rapidly be amortized when solving several linear systems with multiple right-hand sides.

An important aspect of our approach is that the proposed CHEBFILTERCG algorithm requires only matrix-vector products plus some vector updates and many fewer dot products, since these do not appear in the Chebyshev steps that are performed at each conjugate gradient iteration during the solution of the first system. This is of some importance in the context of parallel computing and, in particular, in distributed memory environments where the computation of the dot product requires particular attention.

Our perspectives are to investigate also this approach to construct multilevel preconditioners combining Chebyshev filters and a Krylov basis generated by CHEBFILTERCG when solving the first system of the sequence. In the case of the resolution of nonlinear problems, it is possible to benefit simply from the low intrinsic dimension of these Krylov bases to construct an adaptive preconditioner based on the Krylov space information generated at previous steps in the nonlinear iteration (cf. [20, 29]).

**Acknowledgments.** We thank Iain Duff and Gérard Meurant for their constructive comments and fruitful discussions during the development of this work and also for their careful reading and remarks which were very helpful to improve the paper.

## REFERENCES

[1] M. ARIOLI, I. DUFF, J. NOAILLES, AND D. RUIZ, *A block projection method for sparse matrices*, SIAM J. Sci. Comput., 13 (1992), pp. 47–70.

[2] M. ARIOLI AND D. RUIZ, *A Chebyshev-Based Two-Stage Iterative Method as an Alternative to the Direct Solution of Linear Systems*, Technical report RAL-TR-2002-021, Rutherford Appleton Laboratory, Atlas Center, Didcot, Oxfordshire, OX11 0QX, England, 2002.

[3] S.F. ASHBY, T.A. MANTEUFFEL, AND J.S. OTTO, *A comparison of adaptive Chebyshev and least squares polynomial preconditioning for Hermitian positive definite linear systems*, SIAM J. Sci. Comput., 13 (1992), pp. 1–29.

[4] R. BRAMLEY AND A. SAMEH, *Row projection methods for large nonsymmetric linear systems*, SIAM J. Sci. Comput., 13 (1992), pp. 168–193.

[5] B. CARPENTIERI, L. GIRAUD, AND S. GRATTON, *Additive and multiplicative two-level spectral preconditioning for general linear systems*, SIAM J. Sci. Comput., to appear.

[6] T.F. CHAN AND M.K. NG, *Galerkin projection methods for solving multiple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.

[7] T.F CHAN AND W.L. WAN, *Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Sci. Comput., 18 (1997), pp. 1698–1721.

[8] P. CONCUS, G.H. GOLUB, AND D.P. O'LEARY, *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations*, in Proceedings of the Symposium on Sparse Matrix Computations, Argonne National Laboratory, 1975, Academic, New York, 1976, pp. 309–332.

[9] T. ELFVING, *Block-iterative methods for consistent and inconsistent linear equations*, Numer. Math., 35 (1980), pp. 1–12.

[10] J. Erhel and F. Guyomarc'h, *An augmented conjugate gradient method for solving consecutive symmetric positive linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1279–1299.

[11] P.F. Fischer, *Projection techniques for iterative solution of $A\underline{x} = \underline{b}$ with successive right-hand sides*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 193–204.

[12] L. Giraud, D. Ruiz, and A. Touhami, *A comparative study of iterative solvers exploiting spectral information for SPD systems*, SIAM J. Sci. Comput., 27 (2006), pp. 1760–1786.

[13] L. Giraud, D. Ruiz, and A. Touhami, *Krylov based and polynomial iterative solvers combined with partial spectral factorization for SPD linear systems*, in Vector and Parallel Processing, Lecture Notes in Comput. Sci. 3402, M. Daydé, J. Dongarra, and V. Hernández, and J.M.L.M. Palma, eds., Springer-Verlag, Berlin, Heidelberg, 2005, pp. 635–655.

[14] G.H. Golub and M.D. Kent, *Estimates of eigenvalues for iterative methods*, Math. Comput., 53 (1989), pp. 249–263.

[15] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, 3d ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[16] L.A. Hageman and D.M. Young, *Applied Iterative Methods*, Academic Press, New York and London, 1981.

[17] M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–435.

[18] O.G. Johnson, C.A. Micchelli, and G. Paul, *Polynomial preconditioners for conjugate gradient calculations*, SIAM J. Numer. Anal., 20 (1983), pp. 362–376.

[19] R.B. Lehoucq, D.C. Sorensen, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Problem with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.

[20] D. Loghin, D. Ruiz, and A. Touhami, *Adaptive preconditioners for nonlinear systems of equations*, J. Comput. Appl. Math., 189 (2006), pp. 362–374.

[21] D.P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.

[22] D.P. O'Leary, *Yet another polynomial preconditioner for the conjugate gradient algorithm*, Linear Algebra Appl., 154 (1991), pp. 388–377.

[23] B.N. Parlett, *A new look at the Lanczos algorithm for solving symmetric systems of linear equations*, Linear Algebra Appl., 29 (1980), pp. 323–346.

[24] Y. Saad, *Practical use of polynomial preconditionings for the conjugate gradient method*, SIAM J. Sci. Comput., 6 (1985), pp. 865–881.

[25] Y. Saad, *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, Math. Comp., 178 (1987), pp. 651–662.

[26] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.

[27] V. Simoncini and E. Gallopoulos, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.

[28] G. Szegö, *Orthogonal Polynomials*, American Mathematical Society, Providence, RI, 1975.

[29] A. Touhami, *Utilisation des Filtres de Tchebycheff et Construction de Préconditonneurs Spectraux pour l'Accélération des Méthodes de Krylov*, Ph.D. thesis, INPT–ENSEEIHT, Toulouse, France, 2005.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 796–825

# TWO VARIABLE ORTHOGONAL POLYNOMIALS ON THE BICIRCLE AND STRUCTURED MATRICES*

JEFFREY S. GERONIMO[†] AND HUGO WOERDEMAN[‡]

**Abstract.** We consider bivariate polynomials orthogonal on the bicircle with respect to a positive linear functional. The lexicographical and reverse lexicographical orderings are used to order the monomials. Recurrence formulas are derived between the polynomials of different degrees. These formulas link the orthogonal polynomials constructed using the lexicographical ordering with those constructed using the reverse lexicographical ordering. Relations between the coefficients in the recurrence formulas are derived and used to give necessary and sufficient conditions for the existence of a positive linear functional. These results are then used to construct a class of two variable measures supported on the bicircle that are given by one over the magnitude squared of a stable polynomial. Applications to Fejér–Riesz factorization are also given.

**Key words.** bivariate orthogonal polynomials, positive definite linear functionals, moment problem, doubly Toeplitz matrices, recurrence coefficients

**AMS subject classifications.** 42C05, 30E05, 47A57, 15A48, 47B35

**DOI.** 10.1137/060662472

## 1. Introduction.

Bivariate polynomials orthogonal on the bicircle have been investigated mostly in the electrical engineering community in relation to the design of stable recursive filters for two-dimensional filtering. In particular we note the work of Genin and Kamp [7] who were interested in the following problem. Given any two variable polynomials $q(z,w)$, with $q(0,0) \neq 0$, let $a_{k,l}(z,w)$ be its planar least squares inverse polynomial of degree $(k,l)$; i.e., $a_{k,l}$ minimizes the mean quadratic value of $1 - a_{k,l}q$ on the bicircle. What properties does $a_{k,l}$ have? At the time it was conjectured the minimizing polynomials were stable, i.e., $a_{k,l}(z,w) \neq 0,\ |z| \leq 1,\ |w| \leq 1$, which they showed was false. Their investigation was carried further by Delsarte, Genin, and Kamp [4] who developed the connection between these polynomials and matrix polynomials orthogonal on the unit circle [3]. In the development of this connection these authors were led to examine moment matrices that were block Toeplitz matrices where each block entry is itself a Toeplitz matrix. Such structured matrices are called doubly Toeplitz matrices and arise naturally in the bivariate trigonometric moment problem. These types of matrices arose more recently in the work of Geronimo and Woerdeman [8] in their investigation of the bivariate Fejér–Riesz factorization theorem. These authors were able to resolve the question when a strictly positive bivariate trigonometric polynomial of a certain degree can be written as the magnitude squared of a stable polynomial of the same degree. In this work the authors used the fact that the theory of orthogonal polynomials on the unit circle provides a proof of the one variable Fejér–Riesz theorem which does not use the fundamental theorem of algebra. We intend here to continue to investigate the properties of bivariate polynomials orthogonal on the bicircle and clarify their role in the Fejér–Riesz theorem.

---

[*] Received by the editors June 8, 2006; accepted for publication (in revised form) by D. A. Bini February 23, 2007; published electronically July 11, 2007. The authors were partially supported by an NSF grant and a NATO collaborative linkage grant.

http://www.siam.org/journals/simax/29-3/66247.html

[†] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160 (geronimo@math.gatech.edu).

[‡] Department of Mathematics, Drexel University, Philadelphia, PA 19104 (Hugo.Woerdeman@drexel.edu).

A major difficulty encountered in the theory of orthogonal polynomials of more than one variable is which monomial ordering to use. For bivariate real orthogonal polynomials the preferred ordering is the total degree ordering which is the one set by Jackson [14]. For polynomials with the same total degree the ordering is lexicographical. As noted in Delgado et al. [2] in their study of orthogonal polynomials associated with doubly Hankel matrices, there is a good reason for choosing this ordering which is that if new orthogonal polynomials of higher degree are to be constructed, then their orthogonality relations will not affect the relations governing the lower degree polynomials. However, in order for the moment matrix to be doubly Toeplitz the monomial orderings that need to be used are lexicographical and reverse lexicographical.

We begin in section 2 by considering finite-dimensional subspaces spanned by the monomials $z^i w^j, |i| \leq n$, $|j| \leq m$, and exhibiting the connection between positive linear functionals defined on this space and positive definite doubly Toeplitz matrices. We then introduce certain matrix orthogonal polynomials and show how they give the Cholesky factors for the inverse of the doubly Toeplitz matrices considered above. The results in [8] show that these polynomials play a role in the parametric moment problem. In section 3 we construct two variable orthogonal polynomials, where the monomials are ordered according to the lexicographical ordering. When these polynomials are organized into vector orthogonal polynomials, they can be related to the matrix orthogonal polynomials constructed previously. From this relation it is shown that these vector polynomials are the minimizers of a certain quadratic functional. Using the orthogonality relation, recurrence relations satisfied by the vector polynomials and their counterparts in the reverse lexicographical ordering are derived, and relations between these recurrence coefficients are exhibited. In section 4 a number of Christoffel–Darboux-like formulas are derived. In section 5 we use the relations between the coefficients derived in section 3 to develop an algorithm to construct the coefficients in the recurrence formulas at a particular level $(n, m)$, say, in terms of the coefficients at the previous levels plus a certain number of unknowns. The collection of these unknowns is in one to one correspondence with the number of moments needed to construct the vector polynomials up to level $(n, m)$. This is used in section 6 to construct a positive linear functional from the recurrence coefficients. The construction allows us to find necessary and sufficient conditions on the recurrence coefficients for the existence of a positive linear functional which is in one to one correspondence with the set of positive definite doubly Toeplitz matrices. In section 7 we examine conditions under which the linear functional can be represented as a positive measure supported on the bicircle having the form of one over the magnitude squared of a stable polynomial. This gives a new proof of the Fejér–Riesz result of [8]. Finally in section 8 examples are given that illustrate various aspects of the theory developed.

## 2. Positive linear functionals and doubly Toeplitz matrices.

In this section we consider moment matrices associated with the lexicographical ordering, which is defined by

$$(k, \ell) <_{\text{lex}} (k_1, \ell_1) \Leftrightarrow k < k_1 \text{ or } (k = k_1 \text{ and } \ell < \ell_1),$$

and the reverse lexicographical ordering, defined by

$$(k, \ell) <_{\text{revlex}} (k_1, \ell_1) \Leftrightarrow (\ell, k) <_{\text{lex}} (\ell_1, k_1).$$

Both of these orderings are linear orders, and in addition they satisfy

$$(k, \ell) < (m, n) \Rightarrow (k + p, \ell + q) < (m + p, n + q).$$

In such a case, one may associate a half-space with the ordering which is defined by $\{(k,l) : (0,0) < (k,l)\}$. In the case of the lexicographical ordering we shall denote the associated half-space by $H$ and refer to it as [illegible]. In the case of the reverse lexicographical ordering we shall denote the associated half-space by $\tilde{H}$. Instead of starting with the ordering, one may also start with a half-space $\hat{H}$ of $\mathbb{Z}^2$ (i.e., a set $\hat{H}$ satisfying $\hat{H} + \hat{H} \subset \hat{H}$, $\hat{H} \cap (-\hat{H}) = \emptyset$, $\hat{H} \cup (-\hat{H}) \cup \{(0,0)\} = \mathbb{Z}^2$) and define an ordering via

$$(k,l) <_{\hat{H}} (k_1,l_1) \Longleftrightarrow (k_1 - k, l_1 - l) \in \hat{H}.$$

We shall refer to the order $<_{\hat{H}}$ as [illegible] $\hat{H}$. Note that the lexicographical and reverse lexicographical orderings do not respect total degree.

Let $\prod^{n,m}$ denote the bivariate Laurent linear subspace span$\{z^i w^j, -n \le i \le n, -m \le j \le m\}$. Let $\mathcal{L}_{n,m}$ be a linear functional defined on $\prod^{n,m}$ by

$$\mathcal{L}_{n,m}(z^{-i}w^{-j}) = c_{i,j} = \overline{\mathcal{L}(z^i w^j)}.$$

We will call $c_{i,j}$ the $(i,j)$ moment of $\mathcal{L}_{n,m}$ and $\mathcal{L}_{n,m}$ a moment functional. If we form the $(n+1)(m+1) \times (n+1)(m+1)$ matrix $C_{n,m}$ for $\mathcal{L}_{n,m}$ in the lexicographical ordering, then, as noted in the introduction, it has the special block Toeplitz form

$$C_{n,m} = \begin{bmatrix} C_0 & C_{-1} & \cdots & C_{-n} \\ C_1 & C_0 & \cdots & C_{-n+1} \\ \vdots & & \ddots & \vdots \\ C_n & C_{n-1} & \cdots & C_0 \end{bmatrix}, \tag{2.1}$$

where each $C_i$ is an $(m+1) \times (m+1)$ Toeplitz matrix as follows:

$$C_i = \begin{bmatrix} c_{i,0} & c_{i,-1} & \cdots & c_{i,-m} \\ \vdots & & \ddots & \vdots \\ c_{i,m} & & \cdots & c_{i,0} \end{bmatrix}, \qquad i = -n, \ldots, n. \tag{2.2}$$

Thus $C_{n,m}$ has a doubly Toeplitz structure. If the reverse lexicographical ordering is used in place of the lexicographical ordering, we obtain another moment matrix $\tilde{C}_{n,m}$ where the roles of $n$ and $m$ are interchanged.

Let us introduce the notion of centrotranspose symmetry. We denote the transpose of a matrix $A$ by $A^T$. A square matrix $A$ is said to be [illegible] if $JAJ = A^T$, where $J$ is the matrix with ones on the antidiagonal and zeros elsewhere. Note that a Toeplitz matrix is centrotranspose symmetric. We have the following useful lemmas which characterize Toeplitz and doubly Toeplitz matrices in terms of centrotranspose symmetry.

LEMMA 2.1. [illegible] $(n+1) \times (n+1)$ [illegible] $A = (a_{i,j})_{i,j=0}^{n}$ [illegible] $A$ [illegible] $\hat{A} := (a_{i,j})_{i,j=0}^{n-1}$ [illegible]

[illegible] Notice that $JAJ = A^T$ is equivalent to $a_{n-i,n-j} = a_{j,i}$, $0 \le i,j \le n$. Similarly, the centrotranspose symmetry of $\hat{A}$ is equivalent to $a_{n-1-i,n-1-j} = a_{j,i}$, $0 \le i,j \le n-1$. But then

$$a_{i+1,j+1} = a_{n-j-1,n-i-1} = a_{i,j}, \quad 0 \le i,j \le n-1,$$

and thus it follows that $A$ is Toeplitz.

As $A$ and $\hat{A}$ are Toeplitz, the converse is immediate. □

LEMMA 2.2. [illegible] $A = (A_{i,j})$ $i,j = 1,\dots,k$ [illegible] $A_{i,j}$ [illegible] $m \times m$ [illegible] $A$ [illegible] $A^T = JAJ$ $A_1^T = J_1A_1J_1$ [illegible] $A_2^T = J_1A_2J_1$ [illegible] $A_1$ [illegible] $A$ [illegible] $A_2$ [illegible] $A$ [illegible] $A_{i,j}$ [illegible] $J$ [illegible] $J_1$ [illegible]

[illegible]. Again the necessary conditions follow from the structure of $A$. To see the converse note that $A^T = JAJ$ implies that $A_{j,i}^T = J_2A_{k-i,k-j}J_2$, where $J_2$ is the $m\times m$ matrix with ones on the reverse diagonal and zeros everywhere else. This coupled with the condition on $A_1$ implies that $A$ is a block Toeplitz matrix from Lemma 2.1 and $J_2A_{i,j}J_2 = A_{i,j}^T$. These relations plus the condition on $A_2$ and Lemma 2.1 give the result. □

[illegible] 2.3. The conclusions of the above lemmas hold if we replace deleting the last (last block) row and column by deleting the first (first block) row and column.

We say that the moment functional $\mathcal{L}_{n,m} : \prod^{n,m} \to \mathbb{C}$ is positive definite or positive semidefinite if

$$\mathcal{L}_{n,m}(|p|^2) > 0 \quad \text{or} \quad \mathcal{L}_{n,m}(|p|^2) \geq 0 \tag{2.3}$$

for every nonzero polynomial $p \in \prod^{n,m}$. It follows from a simple quadratic form argument that $\mathcal{L}_{n,m}$ is positive definite or positive semidefinite if and only if its moment matrix $C_{n,m}$ is positive definite or positive semidefinite, respectively.

We will say that $\mathcal{L}$ is positive definite or positive semidefinite if

$$\mathcal{L}(|p|^2) > 0 \qquad \text{or} \qquad \mathcal{L}(|p|^2) \geq 0$$

for all nonzero polynomials, respectively. Again these conditions are equivalent to the moment matrices $C_{n,m}$ being positive definite or positive semidefinite for all positive integers $n$ and $m$. The above discussion leads to the following.

LEMMA 2.4. [illegible] $C_{n,m}$ [illegible] $(n+1)(m+1) \times (n+1)(m+1)$ [illegible] (2.1) [illegible] (2.2) [illegible] $\mathcal{L}_{n,m} : \prod^{n,m} \to \mathbb{C}$ [illegible] $C_{n,m}$ [illegible]

$$c_{i,j} = \mathcal{L}_{n,m}(z^{-i}w^{-j}) = \overline{\mathcal{L}_{n,m}(z^iw^j)}, \qquad -n \leq i \leq n, \qquad -m \leq j \leq m.$$

[illegible]

Let $\prod_{m+1}^n$ be the set of all $(m+1)\times(m+1)$ complex-valued matrix polynomials of degree $n$ or less, $\prod_{m+1}$ the set of all $(m+1)\times(m+1)$ complex-valued matrix polynomials, and $M^{m,n}$ the space of $m \times n$ matrices. For a matrix $M$ we let $M^\dagger$ denote the conjugate transpose (or the adjoint) of $M$. For a polynomial $Q(z,w)$ we let $Q^\dagger(z,w)$ denote the polynomial in $z^{-1}$ and $w^{-1}$ defined by $Q(z,w)^\dagger = Q^\dagger(\frac{1}{\bar{z}}, \frac{1}{\bar{w}})^\dagger$. If the positive moment functional $\mathcal{L}_{n,m} : \prod^{n,m} \to \mathbb{C}$ is extended to two variable polynomials with matrix coefficients in the obvious way, we can associate it with a positive matrix function $\mathcal{L}_m : \prod_{m+1}^n \times \prod_{m+1}^n \to M^{m+1,m+1}$ defined by

$$[\mathcal{L}_m(P(z), Q(z))]_{i,j} = \mathcal{L}_{n,m}(\big[P(z,w)\ Q^\dagger(z,w)\big]_{i,j}), \quad 1 \leq i,j \leq m+1 \tag{2.4}$$

where

$$P(z,w) = P(z)\begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix} \text{ and } Q(z,w) = Q(z)\begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix}.$$

Equation (2.4) shows that if $\mathcal{L}_{n,m}$ can be represented in terms of a positive measure $\mu$ supported on the bicircle, then for $f$ an $(m+1)\times(m+1)$ matrix function continuous on the unit circle,

$$\mathcal{L}_m(f) = \int_{-\pi}^{\pi} f(\theta)dM_m(\theta),$$

where $M_m$ is the $(m+1)\times(m+1)$ matrix measure given by

$$dM_m(\theta) = \int_{\phi=-\pi}^{\pi} \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix} d\mu(\theta,\phi) \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix}^{\dagger},$$

which shows that $M_m$ is Toeplitz.

Because of the structure of $C_{n,m}$ we can associate with $\mathcal{L}_m$ matrix valued orthogonal polynomials in the following manner [3], [4], [8]. Let $\{R_i^m(z)\}_{i=0}^n$ and $\{L_i^m(z)\}_{i=0}^n$ be $(m+1)\times(m+1)$ complex-valued matrix polynomials given by

$$R_i^m(z) = R_{i,i}^m z^i + R_{i,i-1}^m z^{i-1} + \cdots, \qquad i = 0,\ldots,n, \tag{2.5}$$

and

$$L_i^m(z) = L_{i,i}^m z^i + L_{i,i-1}^m z^{i-1} + \cdots, \qquad i = 0,\ldots,n, \tag{2.6}$$

satisfying

$$\mathcal{L}_m(R_i^{m\dagger}, R_j^{m\dagger}) = \delta_{ij} I_{m+1} \tag{2.7}$$

and

$$\mathcal{L}_m(L_i^m, L_j^m) = \delta_{ij} I_{m+1}, \tag{2.8}$$

respectively, where $I_{m+1}$ denotes the $(m+1)\times(m+1)$ identity matrix. The above relations uniquely determine the sequences $\{R_i^m\}_{i=0}^n$ and $\{L_i^m\}_{i=0}^n$ up to a unitary factor, and this factor will be fixed by requiring $R_{i,i}^m$ and $L_{i,i}^m$ to be upper triangular matrices with positive diagonal entries. We write

$$L_i^m(z) = [0 \cdots 0 \ L_{i,i}^m \ L_{i,i-1}^m \ \cdots \ L_{i,0}^m] \begin{bmatrix} z^n I_{m+1} \\ z^{n-1} I_{m+1} \\ \vdots \\ I_{m+1} \end{bmatrix}, \tag{2.9}$$

and

$$\hat{L}_n^m(z) = \begin{bmatrix} L_n^m(z) \\ L_{n-1}^m(z) \\ \vdots \\ L_0^m(z) \end{bmatrix} = L \begin{bmatrix} z^n I_{m+1} \\ z^{n-1} I_{m+1} \\ \vdots \\ I_{m+1} \end{bmatrix}, \tag{2.10}$$

where

$$L = \begin{bmatrix} L_{n,n}^m & L_{n,n-1}^m & \cdots & L_{n,0}^m \\ 0 & L_{n-1,n-1}^m & \cdots & L_{n-1,0}^m \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & L_{0,0}^m \end{bmatrix}. \tag{2.11}$$

In an analogous fashion write

$$\hat{R}_n^m(z) = \begin{bmatrix} R_0^m(z) \\ R_1^m(z) \\ \vdots \\ R_n^m(z) \end{bmatrix} = \begin{bmatrix} I_{m+1} & \dots & z^n I_{m+1} \end{bmatrix} R, \tag{2.12}$$

where

$$R = \begin{bmatrix} R_{0,0}^m & R_{1,0}^m & \cdots & R_{n,0}^m \\ 0 & R_{1,1}^m & \cdots & R_{n,1}^m \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & R_{n,n}^m \end{bmatrix}. \tag{2.13}$$

By lower (respectively, upper) Cholesky factor $A$ (respectively, $B$) of a positive definite matrix $M$, we mean

$$M = AA^\dagger = BB^\dagger, \tag{2.14}$$

where $A$ is a lower triangular matrix with positive diagonal elements, and $B$ is an upper triangular matrix with positive diagonal elements. With the above we have the following well known lemma [15].

Lemma 2.5. [illegible] $C_{n,m}$ [illegible] (2.1). [illegible] $L^\dagger$ [illegible] $R$ [illegible] $C_{n,m}^{-1}$ [illegible]. To obtain (2.14) note that (2.8) implies that

$$I = \mathcal{L}_m(\hat{L}_n^m, \hat{L}_n^m) = L\mathcal{L}_m\left( \begin{bmatrix} z^n I_{m+1} \\ z^{n-1} I_{m+1} \\ \vdots \\ I_{m+1} \end{bmatrix}, \begin{bmatrix} z^n I_{m+1} \\ z^{n-1} I_{m+1} \\ \vdots \\ I_{m+1} \end{bmatrix} \right) L^\dagger = LC_{n,m}L^\dagger,$$

where $I$ is the $(n+1)(m+1) \times (n+1)(m+1)$ identity matrix. Since $C_{n,m}$ is invertible we find

$$C_{n,m}^{-1} = L^\dagger L.$$

The result for $R$ follows in an analogous manner. □

From this formula and (2.11) we find

$$L_n^m(z) = \left[ ({L_{n,n}^m}^\dagger)^{-1}, 0, 0, \dots 0 \right] C_{n,m}^{-1} [z^n I_{m+1}, z^{n-1} I_{m+1}, \dots, I_{m+1}]^T, \tag{2.15}$$

and

$$R_n^m(z) = [I_{m+1}, zI_{m+1}, \dots, z^n I_{m+1}]\, C_{n,m}^{-1} \left[0, 0, \dots, 0, (\bar{R}_{n,n}^m)^{-1}\right]^T. \tag{2.16}$$

Note that ${L_{n,n}^m}^\dagger$ is the lower Cholesky factor of $[I_{m+1}, 0, \cdots, 0] C_{n,m}^{-1} \, [I_{m+1}, 0, \cdots, 0]^T$, while $R_{n,n}^m$ is the upper Cholesky factor of $[0, \cdots, I_{m+1}] C_{n,m}^{-1} \, [0, \cdots, I_{m+1}]^T$.

The theory of matrix orthogonal polynomials (see [3], [15], [17], [19]) can be applied to obtain the recurrence formulas

$$\begin{aligned} A_{i+1,m} L_{i+1}^m(z) &= zL_i^m(z) - E_{i+1,m} \overleftarrow{R}_i^m(z), \quad i = 0, \dots, n-1, \\ R_{i+1}^m(z)\hat{A}_{i+1,m} &= zR_i^m(z) - \overleftarrow{L}_i^m(z) E_{i+1,m}, \quad i = 0, \dots, n-1, \end{aligned} \tag{2.17}$$

where

$$E_{i+1,m} = \mathcal{L}_m(zL_i^m, \overleftarrow{R}_i^m) = \mathcal{L}_m\big(\overleftarrow{L}_i^{m\,\dagger}, (zR_i^m)^\dagger\big) \tag{2.18}$$

and

$$\begin{aligned} A_{i+1,m} &= \mathcal{L}_m(zL_i^m,\ L_{i+1}^m) = L_{i,i}^m(L_{i+1,i+1}^m)^{-1}, \\ \hat{A}_{i+1,m} &= \mathcal{L}_m\big(R_{i+1}^{m\dagger}, (zR_i^m)^\dagger\big) = (R_{i+1,i+1}^m)^{-1}R_{i,i}^m. \end{aligned} \tag{2.19}$$

For a matrix polynomial $B$ of degree $n$ in $z$, $\overleftarrow{B}(z) = z^n\sum_{i=0}^n B_i^\dagger z^{-i}$. By multiplying the first equation in (2.17) on the left by $\bar{z}L_i^m(z)^\dagger$ and the second equation on the right by $\bar{z}R_i^m(z)^\dagger$ and then integrating, we see that

$$\begin{aligned} A_{i+1,m}A_{i+1,m}^\dagger &= I_{m+1} - E_{i+1,m}E_{i+1,m}^\dagger, \\ \hat{A}_{i+1,m}^\dagger\hat{A}_{i+1,m} &= I_{m+1} - E_{i+1,m}^\dagger E_{i+1,m}. \end{aligned} \tag{2.20}$$

The above equations and the properties of $A_{i+1,m}$ and $\hat{A}_{i+1,m}$ show that $E_{i+1,m}$ is a strictly contractive matrix and that $A_{i+1,m}$ is the upper Cholesky factor of $I_{m+1} - E_{i+1,m}E_{i+1,m}^\dagger$. Similarly $\hat{A}_{i,m}^\dagger$ is the lower Cholesky factor of $I_{m+1} - E_{i+1,m}^\dagger E_{i+1,m}$. Furthermore (2.19) and (2.20) show that

$$\det((L_{i+1,i+1}^m)^\dagger L_{i+1,i+1}^m)^{-1} = \det(C_0)\prod_{j=1}^{i+1}\det(I_{m+1} - E_{j,m}E_{j,m}^\dagger). \tag{2.21}$$

The recurrence formulas (2.17) can be inverted in the following manner. Multiply the reverse of the second equation in (2.17) on the right by $E_{i+1,m}$ to obtain

$$E_{i+1,m}\hat{A}_{i+1,m}^\dagger\overleftarrow{R}_{i+1}^m(z) = E_{i+1,m}\overleftarrow{R}_i^m(z) - zE_{i+1,m}E_{i+1,m}^\dagger L_i^m(z).$$

Add this equation to the first equation in (2.17) and then use (2.20) to eliminate $A_{i+1,m}$ and $\hat{A}_{i+1,m}^\dagger$ to find

$$(A_{i+1,m}^\dagger)^{-1}L_{i+1}^m(z) + E_{i+1,m}(\hat{A}_{i+1,m})^{-1}\overleftarrow{R}_{i+1}^m(z) = zL_i^m(z). \tag{2.22}$$

In a similar manner we find

$$R_{i+1}^m(z)(\hat{A}_{i+1,m}^\dagger)^{-1} + \overleftarrow{L}_{i+1}^m(z)(A_{i+1,m})^{-1}E_{i+1,m} = zR_i^m(z). \tag{2.23}$$

From the recurrence formulas it is not difficult to derive the Christoffel–Darboux formulas [3]:

$$\begin{aligned} \overleftarrow{R}_k^m(z)^\dagger\overleftarrow{R}_k^m(z_1) - \bar{z}z_1L_k^m(z)^\dagger L_k^m(z_1) &= (1-\bar{z}z_1)\sum_{i=0}^k L_i^m(z)^\dagger L_i^m(z_1), \\ \overleftarrow{L}_k^m(z_1)\overleftarrow{L}_k^m(z)^\dagger - \bar{z}z_1R_k^m(z_1)R_k^m(z)^\dagger &= (1-\bar{z}z_1)\sum_{i=0}^k R_i^m(z_1)R_i^m(z)^\dagger. \end{aligned} \tag{2.24}$$

These formulas give rise to the matrix Gohberg–Semencul formulas [11], [15] when the linear equations obtained by equating like powers of $\bar{z}^iz_1^j$ are put in matrix form.

Some properties that follow from the above formulas [3, Theorems 9, 14, and 15] are that $\overleftarrow{R}_i^m(z)$ and $\overleftarrow{L}_k^m(z)$ have empty kernels for $|z| \le 1$; i.e.,

$$\det(\overleftarrow{R}_i^m(z)) \ne 0 \ne \det(\overleftarrow{L}_k^m(z)), \ |z| \le 1. \tag{2.25}$$

Such polynomials are called stable matrix polynomials, and if we write

$$W_k(z) = \left[\overleftarrow{L}_k^m(z)\overleftarrow{L}_k^m(z)^\dagger\right]^{-1} \tag{2.26}$$

and

$$C_j^k = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-ij\theta} W_k(e^{i\theta})d\theta,$$

then

$$C_j^k = C_j, \quad |j| \le k. \tag{2.27}$$

Furthermore

$$W_k = \left[\overleftarrow{R}_k^m(z)^\dagger \overleftarrow{R}_k^m(z)\right]^{-1}. \tag{2.28}$$

If $\overleftarrow{L}_k^m(z)$ ($\overleftarrow{R}_k^m(z)$) satisfies (2.25) and (2.27), we will say it is stable and has spectral matching (up to level $k$). Another useful result shown in [3] is

$$\log\det((L_{i+1,i+1}^m)^\dagger L_{i+1,i+1}^m)^{-1} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\det W_k(\theta)d\theta. \tag{2.29}$$

From the stability of $\overleftarrow{R}_{i+1}^m$ and $\overleftarrow{L}_{i+1}^m$, (2.22) and (2.23) give the following formulas for the recurrence coefficients $E_{i+1,m}$:

$$\begin{aligned} E_{i+1,m} &= -(A_{i+1,m}^\dagger)^{-1} L_{i+1}^m(0)\overleftarrow{R}_{i+1}^m(0)^{-1}\hat{A}_{i+1,m} \\ &= -A_{i+1,m}\overleftarrow{L}_{i+1}^m(0)^{-1}R_{i+1}^m(0)(\hat{A}_{i+1,m}^\dagger)^{-1}. \end{aligned} \tag{2.30}$$

We also note that $\overleftarrow{L}_k^m(z)$ and $\overleftarrow{R}_k^m(z)$ are minimizers of certain quadratic functions. To see this denote the set of $(m+1)\times(m+1)$ hermitian matrices as $\mathrm{Herm}(m+1)$, and let $\mathcal{M} : \prod_{m+1} \to \mathrm{Herm}(m+1)$ be given by

$$\mathcal{M}[X(z)] = \mathcal{L}_m(X, X) - (X(0) + X(0)^\dagger), \tag{2.31}$$

then Delsarte, Genin, and Kamp have shown [3] that for a given degree $k$, $\mathcal{M}$ is minimized by $\overleftarrow{L}_k^m(z)L_{k,m}^m$ with the value $(L_{k,m}^m)^\dagger L_{k,m}^m$. Likewise $\hat{\mathcal{M}} : \prod_{m+1} \to \mathrm{Herm}(m+1)$ given by

$$\hat{\mathcal{M}}[X(z)] = \mathcal{L}_m(X^\dagger, X^\dagger) - (X(0) + X(0)^\dagger) \tag{2.32}$$

is minimized by $R_{k,m}^m\overleftarrow{R}_k^m(z)$ and takes the value $R_{k,m}^m(R_{k,m}^m)^\dagger$. Thus we find

$$(L_{k,m}^m)^\dagger L_{k,m}^m \ge (L_{k+1,m}^m)^\dagger L_{k+1,m}^m \tag{2.33}$$

and

$$R^m_{k,m}(R^m_{k,m})^\dagger \geq R^m_{k+1,m}(R^m_{k+1,m})^\dagger. \tag{2.34}$$

Here $A \geq B$ for two $(m+1) \times (m+1)$ matrices means that $A - B$ is positive semidefinite. The above discussion leads to Burg's entropy theorem. Consider the class of $M^m$ of $(m+1) \times (m+1)$ matrix Borel measures on the unit circle, and for each such measure $\mu$ write the Lebesgue decomposition of $\mu = \mu_{ac} + \mu_s$, where $d\mu_{ac}/d\theta = W(\theta)$. Let $S^m_n$ be the subset of $M^m$ such that each $\mu \in S^m_n$ has the same Fourier coefficients $C_i$, $|i| \leq n$, and $\mathcal{E}(\mu) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \ln\det(W)d\theta > -\infty$. Then there is a unique measure which maximizes the above entropy function $\mathcal{E}(\mu)$, and this measure is given by $d\mu = W(\theta)d\theta$, with $W(\theta) = Q^m_n(\theta)^{-1}$, where $Q^m_n(\theta)$ is a positive $(m+1) \times (m+1)$ matrix trigonometric polynomial of degree $n$.

This leads to a simple proof of the matrix Fejér–Reisz factorization theorem (Helson [13], Dritschel [5], McLean and Woerdeman [16], Geronimo and Lai [10]) which will be useful later.

LEMMA 2.6. [illegible] $Q^m_n(\theta)$ [illegible] $(m+1)\times(m+1)$ [illegible] $Q^m_n(\theta) = \overleftarrow{L}^m_n(z)(\overleftarrow{L}^m_n(z))^\dagger$, $z = e^{i\theta}$ [illegible] $\overleftarrow{L}^m_n$ [illegible] $(m+1) \times (m+1)$ [illegible] $n$ [illegible] $L^m_n$ [illegible] (2.15)

[illegible] Since $Q^m_n(\theta)$ is strictly positive we can compute the moments $C_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-ij\theta}Q^m_n(\theta)^{-1}d\theta$. If we compute the matrix orthogonal polynomials associated with these Fourier coefficients, we find that $W_n$ has spectral matching up to $n$. That is, its Fourier coefficients match $C_i$ for $|i| \leq n$. The maximum entropy theorem implies that $Q^m_n(\theta) = W_n^{-1}$, which gives the result. □

The matrix Fejér–Riesz theorem now follows.

THEOREM 2.7. [illegible] $Q^m_n(\theta) \geq 0$ [illegible] $(m+1)\times(m+1)$ [illegible] $Q^m_n(\theta) = P^m_n(z)(P^m_n(z))^\dagger$, $z = e^{i\theta}$ [illegible] $P^m_n$ [illegible] $|z| < 1$ $(m+1) \times (m+1)$ [illegible]

[illegible] Let $Q^m_{n,\epsilon} = \epsilon I + Q^m_n$ , $\epsilon > 0$; then $Q^m_{n,\epsilon}$ satisfies the hypotheses of the above lemma. Thus $Q^m_{n,\epsilon} = P^m_{n,\epsilon}(P^m_{n,\epsilon})^\dagger$. The proof now follows by taking the limit as $\epsilon$ tends to zero. □

It was observed by Delsarte et al. [4] that if the $C_k$ in $C_{n,m}$ are centrotranspose symmetric, then

$$(L^{m\dagger}_{i,i}L^m_i(z))^T = J_m R^m_i(z)R^{m\dagger}_{i_i}J_m, \quad i = 0, \ldots, n, \tag{2.35}$$

where $J_m$ is the $(m+1) \times (m+1)$ matrix with ones on the reverse diagonal and zeros everywhere else. This can easily be seen from (2.15) and (2.16) since in this case from Lemma 2.2 $C^T_{n,m} = JC_{n,m}J$, with $J$ the $(n+1)(m+1) \times (m+1)(n+1)$ matrix with ones down the antidiagonal and zeros everywhere else. This leads to the following characterization of positive definite doubly Toeplitz matrices in terms of certain recurrence coefficients. We will denote by $C^m_0$ the $m \times m$ matrix obtained from $C_0$ by eliminating the first row and first column of $C_0$.

THEOREM 2.8. [illegible] $C_{n,m}$ [illegible] $C_i, |i| \leq n$ [illegible] $E_{k,m}, k = 1, \ldots, n$ [illegible] $C_0$ [illegible] $C_{n,m}$ [illegible] $E_{k,i}, k = 1, \ldots, n, i = m-1, m$ $C_0$ [illegible] $C^m_0$ [illegible]

[illegible] Examining the leading coefficients in (2.35) and using the fact that $L^m_{i,i}$ and $R^m_{i,i}$ are upper triangular, we find that (see also [4]) $(L^m_{i,i})^T = J_m R^m_{i,i}J_m$ for $i = 0, \ldots, n$. Thus

$$L^m_i(z)^T = J_m R^m_i(z)J_m, \; i = 0, \ldots, n. \tag{2.36}$$

The above equation and (2.19) imply that

$$J_m A_{i+1,m} J_m = A_{i+1,m}^T. \tag{2.37}$$

Using this coupled with (2.36) and its reverse in (2.30) yields

$$\begin{aligned} J_m E_{i+1,m} J_m &= -J_m (A_{i+1,m}^\dagger)^{-1} L_{i+1}^m(0) \overleftarrow{R}_{i+1}^m(0)^{-1} \hat{A}_{i+1,m} J_m \\ &= -(A_{i+1,m} \overleftarrow{L}_{i+1}^m(0)^{-1} R_{i+1}^m(0) (\hat{A}_{i+1,m}^\dagger)^{-1})^T = E_{i+1,m}^T. \end{aligned} \tag{2.38}$$

To show the converse note that if $E_{i,m}$ is centrotranspose symmetric, then from (2.20) we obtain

$$J_m (A_{i,m} A_{i,m}^\dagger)^T J_m = J_m (I_m - E_{i,m} E_{i,m}^\dagger) J_m = \overline{(I - E_{i,m}^\dagger E_{i,m})} = \hat{A}_{i,m}^T \overline{\hat{A}_{i,m}},$$

which gives (2.37). Since $C_0$ is centrotranspose symmetric and $L_{0,m}^\dagger(z)$ is the lower Cholesky factor of $C_0$, we see that $J_m L_0^m J_m = R_0^{mT}$. Thus by induction using (2.17) we find that $J_m L_n^m(z) J_m = R_n^m(z)^\top$. The first part of the result now follows from the spectral matching of $W_n$, (2.26), and (2.28). The second part of the theorem follows by applying the above argument to $C_{n,m-1}$ and $C_0^m$ and then using Lemma 2.1. ☐

In the next two sections we present recurrence formulas and an algorithm that computes recurrence coefficients for a positive definite doubly Toeplitz matrix.

**3. Bivariate orthogonal polynomials.** In this section we examine the properties of two variable orthogonal polynomials where the monomial ordering is either lexicographical or reverse lexicographical. The study of orthogonal polynomials on the bicircle with this ordering was begun by Delsarte et al. [4] and extended in [8]. Given a positive definite linear functional $\mathcal{L}_{N,M} : \prod^{N,M} \to \mathbb{C}$ we perform the Gram–Schmidt procedure using the lexicographical ordering and define the orthonormal polynomials $\phi_{n,m}^l(z,w)$, $0 \le n \le N$, $0 \le m \le M$, $0 \le l \le m$, by the equations

$$\begin{aligned} &\mathcal{L}_{N,M}(\phi_{n,m}^l z^{-i} w^{-j}) = 0, \quad 0 \le i < n \text{ and } 0 \le j \le m \quad \text{ or } i = n \text{ and } 0 \le j < l, \\ &\mathcal{L}_{N,M}(\phi_{n,m}^l (\phi_{n,m}^l)^\dagger) = 1, \end{aligned} \tag{3.1}$$

and

$$\phi_{n,m}^l(z,w) = k_{n,m,l}^{n,l} z^n w^l + \sum_{(i,j) <_{\text{lex}} (n,l)} k_{n,m,l}^{i,j} z^i w^j. \tag{3.2}$$

With the convention $k_{n,m,l}^{n,l} > 0$, the above equations uniquely specify $\phi_{n,m}^l$. Polynomials orthonormal with respect to $\mathcal{L}_{N,M}$ but using the reverse lexicographical ordering will be denoted by $\tilde{\phi}_{n,m}^l$. They are uniquely determined by the above relations with the roles of $n$ and $m$ interchanged.

Set

$$\Phi_{n,m} = \begin{bmatrix} \phi_{n,m}^m \\ \phi_{n,m}^{m-1} \\ \vdots \\ \phi_{n,m}^0 \end{bmatrix} = K_{n,m} \begin{bmatrix} z^n w^m \\ z^n w^{m-1} \\ \vdots \\ 1 \end{bmatrix}, \tag{3.3}$$

where the $(m+1) \times (n+1)(m+1)$ matrix $K_{n,m}$ is given by

$$K_{n,m} = \begin{bmatrix} k_{n,m,m}^{n,m} & k_{n,m,m}^{n,m-1} & \cdots & \cdots & \cdots & k_{n,m,m}^{0,0} \\ 0 & k_{n,m,m-1}^{n,m-1} & \cdots & \cdots & \cdots & k_{n,m,m-1}^{0,0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & k_{n,m,0}^{n,0} & k_{n,m,0}^{n-1,m} & \cdots & k_{n,m,0}^{0,0} \end{bmatrix}. \tag{3.4}$$

As indicated above denote

$$\tilde{\Phi}_{n,m} = \begin{bmatrix} \tilde{\phi}^{n}_{n,m} \\ \tilde{\phi}^{n-1}_{n,m} \\ \vdots \\ \tilde{\phi}^{0}_{n,m} \end{bmatrix} = \tilde{K}_{n,m} \begin{bmatrix} w^m z^n \\ w^m z^{n-1} \\ \vdots \\ 1 \end{bmatrix}, \tag{3.5}$$

where the $(n+1) \times (n+1)(m+1)$ matrix $\tilde{K}_{n,m}$ is given similarly to (3.4) with the roles of $n$ and $m$ interchanged. For the bivariate polynomials $\phi^l_{n,m}(z,w)$ above we define the reverse polynomials $\overleftarrow{\phi}^{l}_{n,m}(z,w)$ by the relation

$$\overleftarrow{\phi}^{l}_{n,m}(z,w) = z^n w^m \bar{\phi}^l_{n,m}(1/z, 1/w). \tag{3.6}$$

With this definition $\overleftarrow{\phi}^{l}_{n,m}(z,w)$ is again a polynomial in $z$ and $w$, and furthermore

$$\overleftarrow{\Phi}_{n,m}(z,w) := \begin{bmatrix} \overleftarrow{\phi}^{m}_{n,m} \\ \overleftarrow{\phi}^{m-1}_{n,m} \\ \vdots \\ \overleftarrow{\phi}^{0}_{n,m} \end{bmatrix}^T . \tag{3.7}$$

An analogous procedure is used to define $\overleftarrow{\tilde{\phi}}^{l}_{n,m}$.

In order to ease the notation to find recurrence formulas for the vector polynomials $\Phi_{n,m}$, we introduce the inner product

$$\langle X, Y \rangle = \mathcal{L}_{N,M}(XY^\dagger). \tag{3.8}$$

Let $\hat{\prod}^{n,m}$ be the linear span of $z^i w^j$, $0 \le i \le n, 0 \le j \le m$, $\hat{\prod}^{n,m}_k$ be the vector space of $k$ dimensional vectors with entries in $\hat{\prod}^{n,m}$, and $\hat{\prod}^{m}_{m+1} = \hat{\prod}^{\infty,m}_{m+1}$.

Utilizing the orthogonality relations (3.1) we obtain the following auxiliary results.

Lemma 3.1. [illegible] $\Phi \in \hat{\prod}^{n,m}_k$ [illegible] $\Phi$ [illegible]

$$\langle \Phi, z^i w^j \rangle = 0, \quad 0 \le i < n, \quad 0 \le j \le m, \tag{3.9}$$

[illegible] $\Phi = T\Phi_{n,m}$ [illegible] $T$ [illegible] $k \times (m+1)$ [illegible] $k = m+1$ $T$ [illegible] $\langle \Phi, \Phi \rangle = I_{m+1}$ [illegible] $T = I_{m+1}$

Lemma 3.2. [illegible] $\tilde{\Phi} \in \hat{\prod}^{n,m}_k$ [illegible] $\tilde{\Phi}$ [illegible]

$$\langle \tilde{\Phi}, z^i w^j \rangle = 0, \quad 0 \le i \le n, \quad 0 \le j < m, \tag{3.10}$$

[illegible] $\tilde{\Phi} = T\tilde{\Phi}_{n,m}$ [illegible] $T$ [illegible] $k \times (n+1)$ [illegible] $k = n+1$ $T$ [illegible] $\langle \tilde{\Phi}, \tilde{\Phi} \rangle = I_{n+1}$ [illegible] $T = I_{n+1}$

With the above we can make contact with the matrix orthogonal polynomials introduced in section 2. This was observed by Delsarte et al. [4].

Lemma 3.3. [illegible] $\Phi_{n,m}$ [illegible] (3.3) [illegible]

$$\Phi_{n,m} = L^m_n(z)[w^m, w^{m-1}, \ldots, 1]^T, \tag{3.11}$$

$$\overleftarrow{\Phi}_{n,m} = [1, w, \ldots, w^m] J_m \overleftarrow{R}^m_n(z)^T J_m, \tag{3.12}$$

[illegible]

$$\begin{bmatrix} \Phi_{n,m}(z,w) \\ \Phi_{n-1,m}(z,w) \\ \vdots \\ \Phi_{0,m}(z,w) \end{bmatrix} = \begin{bmatrix} L_n^m(z) \\ L_{n-1}^m(z) \\ \vdots \\ L_0^m(z) \end{bmatrix} [w^m, w^{m-1}, \dots, 1]^T$$

$$= L \begin{bmatrix} z^n I_{m+1} \\ z^{n-1} I_{m+1} \\ \vdots \\ I_{m+1} \end{bmatrix} [w^m, w^{m-1}, \dots, 1]^T. \tag{3.13}$$

[illegible] If we substitute the equation

$$\Phi_{n,m} = \hat{L}_n(z)[w^m \ \cdots \ 1]^T = \sum_i \hat{L}_{n,i} z^i [w^m \ \cdots \ 1]^T$$

into (3.9), where $\hat{L}_n(z)$ is an $(m+1) \times (m+1)$ matrix polynomial of degree $n$, we find, for $j = 0, \dots, n-1$,

$$\begin{aligned} 0 &= \left\langle \Phi_{n,m},\, z^j \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix} \right\rangle = \sum_{i=0}^{n} \hat{L}_{n,i} \left\langle z^i \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix},\, z^j \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix} \right\rangle \\ &= \sum_{i=1}^{n} \hat{L}_{n,i} \begin{bmatrix} \mathcal{L}_{NM}(z^{i-j}) & \cdots & \mathcal{L}_{NM}(z^{i-j} w^{-m}) \\ \vdots & & \vdots \\ \mathcal{L}_{NM}(z^{i-j} w^m) & \cdots & \mathcal{L}_{NM}(z^{i-j}) \end{bmatrix} \\ &= \sum_{i=1}^{n} \hat{L}_{n,i} \mathcal{L}_m(z^i,\, z^j) = \mathcal{L}_m(\hat{L}_n(z),\, z^j). \end{aligned}$$

Similarly,

$$\langle \Phi_{n,m},\, \Phi_{n,m} \rangle = I_{m+1} = \mathcal{L}_m \langle \hat{L}_n(z),\, \hat{L}_n(z) \rangle.$$

This, coupled with (2.8) and the fact that (3.3) implies that $\hat{L}_{n,m}$ is upper triangular with positive diagonal entries, gives (3.11). Equation (3.12) follows from (3.11) and (2.36), while (3.13) follows from (3.11) and the definition of $L$. □

Analogous formulas for bivariate orthogonal polynomials in the reverse lexicographical ordering are obtained by interchanging the roles on $n$ and $m$.

The function $\mathcal{M}$ given by (2.31) can be used to show that $\overleftarrow{\Phi}_{n,m}$ satisfies a minimization condition. Define $\bar{\mathcal{M}} : \hat{\prod}_{m+1}^{m} \to \operatorname{Herm}(m+1)$ by

$$\bar{\mathcal{M}}(\Phi) = \langle \Phi^\dagger,\, \Phi^\dagger \rangle - (\Phi_0 + \Phi_0^\dagger).$$

We find the following.

Lemma 3.4. [illegible] $\overleftarrow{\Phi}_{n,m}$ [illegible] $\hat{\prod}_{m+1}^{n,m}$

[illegible] Since $\Phi \in \hat{\prod}_{m+1}^{n,m}$ can be represented as

$$\Phi(z,w) = [1, w, \dots, w^m] \hat{\Phi}(z) = [1, w, \dots, w^m] \sum_{i=0}^{n} \Phi_i z^i,$$

and from (2.4)

$$\left\langle z^i \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix}, z^j \begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix} \right\rangle = \begin{bmatrix} \mathcal{L}_{NM}(z^{i-j}) & \cdots & \mathcal{L}_{NM}(z^{i-j}w^{-m}) \\ \vdots & & \vdots \\ \mathcal{L}_{NM}(z^{i-j}w^{m}) & \cdots & \mathcal{L}_{NM}(z^{i-j}) \end{bmatrix} = \mathcal{L}_m(z^i, z^j),$$

we find $\bar{\mathcal{M}}(\Phi) = \hat{\mathcal{M}}(\hat{\Phi})$. The result now follows from (3.12) and the fact that $R^m_{n,m}\overleftarrow{R}^m_n(z)$ minimizes $\hat{\mathcal{M}}$ on $\prod^n_{m+1}$. ∎

We can now derive recurrence relations between the various polynomials.

Theorem 3.5. [illegible] $\{\Phi_{n,m}\}$ [illegible] $\{\tilde{\Phi}_{n,m}\}$ $0 \le n \le N$ $0 \le m \le M$ [illegible]

$$A_{n,m}\Phi_{n,m} = z\Phi_{n-1,m} - \hat{E}_{n,m}\overleftarrow{\Phi}^T_{n-1,m}, \tag{3.14}$$

$$\Phi_{n,m} + A^\dagger_{n,m}\hat{E}_{n,m}(A^T_{n,m})^{-1}\overleftarrow{\Phi}^T_{n,m} = A^\dagger_{n,m}z\Phi_{n-1,m}, \tag{3.15}$$

$$\Gamma_{n,m}\Phi_{n,m} = \Phi_{n,m-1} - \mathcal{K}_{n,m}\tilde{\Phi}_{n-1,m}, \tag{3.16}$$

$$\Gamma^1_{n,m}\Phi_{n,m} = w\Phi_{n,m-1} - \mathcal{K}^1_{n,m}\overleftarrow{\tilde{\Phi}}^T_{n-1,m}, \tag{3.17}$$

$$\Phi_{n,m} = I_{n,m}\tilde{\Phi}_{n,m} + \Gamma^\dagger_{n,m}\Phi_{n,m-1}, \tag{3.18}$$

$$\overleftarrow{\Phi}^T_{n,m} = I^1_{n,m}\tilde{\Phi}_{n,m} + (\Gamma^1_{n,m})^T\overleftarrow{\Phi}^T_{n,m-1}, \tag{3.19}$$

[illegible]

$$\hat{E}_{n,m} = \langle z\Phi_{n-1,m}, \overleftarrow{\Phi}^T_{n-1,m}\rangle = E_{n,m}J_m = \hat{E}^T_{n,m} \in M^{m+1,m+1}, \tag{3.20}$$

$$A_{n,m} = \langle z\Phi_{n-1,m}, \Phi_{n,m}\rangle \in M^{m+1,m+1}, \tag{3.21}$$

$$\mathcal{K}_{n,m} = \langle \Phi_{n,m-1}, \tilde{\Phi}_{n-1,m}\rangle \in M^{m,n}, \tag{3.22}$$

$$\Gamma_{n,m} = \langle \Phi_{n,m-1}, \Phi_{n,m}\rangle \in M^{m,m+1}, \tag{3.23}$$

$$\mathcal{K}^1_{n,m} = \langle w\Phi_{n,m-1}, \overleftarrow{\tilde{\Phi}}^T_{n-1,m}\rangle \in M^{m,n}, \tag{3.24}$$

$$\Gamma^1_{n,m} = \langle w\Phi_{n,m-1}, \Phi_{n,m}\rangle \in M^{m,m+1}, \tag{3.25}$$

$$I_{n,m} = \langle \Phi_{n,m}, \tilde{\Phi}_{n,m}\rangle \in M^{m+1,n+1}, \tag{3.26}$$

$$I^1_{n,m} = \langle \overleftarrow{\Phi}^T_{n,m}, \tilde{\Phi}_{n,m}\rangle \in M^{m+1,n+1}. \tag{3.27}$$

[illegible] 3.6. Formulas similar to (3.14)–(3.19) hold for $\tilde{\Phi}_{n,m}$ and will be denoted by $(\tilde{3}.14)$–$(\tilde{3}.19)$. Throughout the rest of the paper we use the same notation to denote the extension to $\tilde{\Phi}_{n,m}$ of existing formulas stated for $\Phi_{n,m}$.

[illegible]. Equation (3.14) follows from Lemma 3.3, (2.17), (2.36), and (2.37). Likewise (3.15) follows in an analogous manner from (2.22). To prove (3.16) note that, because of the linear independence of the entries of $\Phi_{n,m}$, there is an $m \times (m+1)$ matrix $\Gamma_{n,m}$ such that $\Gamma_{n,m}\Phi_{n,m} - \Phi_{n,m-1} \in \hat{\prod}^{n-1,m}_m$. Furthermore

$$\langle \Gamma_{n,m}\Phi_{n,m} - \Phi_{n,m-1}, z^i w^j\rangle = 0, \qquad 0 \le i \le n-1, \quad 0 \le j \le m-1.$$

Thus Lemma 3.2 implies that

$$\Gamma_{n,m}\Phi_{n,m} - \Phi_{n,m-1} = H_{n,m}\tilde{\Phi}_{n-1,m}.$$

The remaining recurrence formulas follow in a similar manner. ∎

3.7. As indicated in the proof, (3.14) follows from the theory of matrix orthogonal polynomials and so allows us to compute in the $n$ direction along a strip of size $m+1$. This formula does not mix the polynomials in the two orderings. However, to increase $m$ by one for polynomials constructed in the lexicographical ordering, the remaining relations show that orthogonal polynomials in the reverse lexicographical ordering must be used.

Using the orthogonality relations from Lemmas 3.1 and 3.2 and (3.1), we find the following relations.

PROPOSITION 3.8. $\tilde{\Phi}$ $\Phi$

$$\tilde{\mathcal{K}}_{n,m} = \mathcal{K}^{\dagger}_{n,m}, \ \ \tilde{I}_{n,m} = I^{\dagger}_{n,m}, \tag{3.28}$$

$$\tilde{I}^{1}_{n,m} = (I^{1}_{n,m})^{T}, \ \ \tilde{\mathcal{K}}^{1}_{n,m} = (\mathcal{K}^{1}_{n,m})^{T}. \tag{3.29}$$

$$A_{n,m}A^{\dagger}_{n,m} = I_m - \hat{E}_{n,m}\hat{E}^{\dagger}_{n,m}, \tag{3.30}$$

$$\Gamma_{n,m}\Gamma^{\dagger}_{n,m} = I_m - \mathcal{K}_{n,m}\mathcal{K}^{\dagger}_{n,m}, \tag{3.31}$$

$$\Gamma^{1}_{n,m}(\Gamma^{1}_{n,m})^{\dagger} = I_m - \mathcal{K}^{1}_{n,m}(\mathcal{K}^{1}_{n,m})^{\dagger}, \tag{3.32}$$

$$I_{n,m}I^{\dagger}_{n,m} + \Gamma^{\dagger}_{n,m}\Gamma_{n,m} = I_{m+1}, \tag{3.33}$$

$$I^{1}_{n,m}(I^{1}_{n,m})^{\dagger} + (\Gamma^{1}_{n,m})^{\dagger}\Gamma^{1}_{n,m} = I_{m+1}. \tag{3.34}$$

3.9. The matrix $\Gamma_{n,m}$ has a zero in the entries $(i,j), i \geq j$, and has positive $(i, i+1)$ entries. Since $\Gamma_{n,m}\Gamma^{\dagger}_{n,m} = \Gamma_{n,m}U^{\dagger}_{m}U_{m}\Gamma^{\dagger}_{n,m}$, where $U_m$ is the $m \times m+1$ matrix given by

$$U_m = \begin{bmatrix} 0, & I_m \end{bmatrix}, \tag{3.35}$$

we see that $\Gamma_{n,m}U^{\dagger}_{m}$ is the upper Cholesky factorization of the right-hand side of (3.31). From this $\Gamma_{n,m}$ can be obtained once $\mathcal{K}_{n,m}$ is specified. The matrix $\Gamma^{1}_{n,m}$ has zeros in the entries $(i,j), i > j$, with positive $(i,i)$ entries. The matrix $I_{n,m}$ has the first row and column equal to zero except for a one in the $(1,1)$ entry.

The above recurrence formulas also give pointwise formulas for the recurrence coefficients. In order to obtain these formulas we define the $m \times m+1$ matrix $U^{1}_{m}$ as

$$U^{1}_{m} = \begin{bmatrix} I_m, & 0 \end{bmatrix}, \tag{3.36}$$

and the $(n+1)(m+1) \times (n+1)(m+1)$ matrix $P^{n,m}_{rl}$, which takes monomials in the lexicographical ordering to those in the reverse lexicographical ordering; i.e.,

$$P^{n,m}_{rl}[z^n w^m, z^n w^{m-1}, \ldots, 1]^T = [w^m z^n, w^m z^{n-1}, \ldots, 1]^T. \tag{3.37}$$

Analogous equations hold for the $n \times (n+1)$ matrices $\tilde{U}_n$ and $\tilde{U}^{1}_{n}$.

PROPOSITION 3.10.

$$\Phi_{n,m}(z,w) = \Phi^{m}_{n}(z)\begin{bmatrix} w^m \\ \vdots \\ 1 \end{bmatrix} \text{ and } \tilde{\Phi}_{n,m}(z,w) = \tilde{\Phi}^{n}_{m}(w)\begin{bmatrix} z^n \\ \vdots \\ 1 \end{bmatrix}, \tag{3.38}$$

$$\Phi^{m}_{n}(z) = \Phi^{m}_{n,n}z^{n} + \Phi^{m}_{n,n-1}z^{n-1} + \cdots,$$

$$\tilde{\Phi}^{n}_{m}(w) = \tilde{\Phi}^{n}_{m,m}w^{m} + \tilde{\Phi}^{n}_{m,m-1}w^{m-1} + \cdots; \tag{3.39}$$

[illegible]

$$\Gamma_{n,m} = \Phi^{m-1}_{n,n} U_m (\Phi^m_{n,n})^{-1}, \tag{3.40}$$

$$\Gamma^1_{n,m} = \Phi^{m-1}_{n,n} U^1_m (\Phi^m_{n,n})^{-1}, \tag{3.41}$$

$$\mathcal{K}_{n,m} = -\Gamma_{n,m} I_{n,m} \tilde{F}_{n,m}, \tag{3.42}$$

$$\mathcal{K}^1_{n,m} = -\Gamma^1_{n,m} \bar{I}^1_{n,m} \bar{\tilde{F}}^1_{n,m}, \tag{3.43}$$

$$I_{n,m} = ({\Phi^m_{n,n}}^{\dagger})^{-1} [I_{m+1}, 0, \dots, 0] C^{-1}_{n,m} {P^{n,m}_{rl}}^T [I_{n+1}, 0, \dots, 0]^T (\tilde{\Phi}^n_{m,m})^{-1}, \tag{3.44}$$

$$I^1_{n,m} = ({\Phi^m_{n,n}}^{T})^{-1} [0, \dots, 0, J_{m+1}] C^{-1}_{n,m} {P^{n,m}_{rl}}^T [I_{n+1}, 0, \dots, 0]^T (\tilde{\Phi}^n_{m,m})^{-1}, \tag{3.45}$$

[illegible] $\tilde{F}_{n,m} = \tilde{\Phi}^n_{m,m} U^T_n (\tilde{\Phi}^{n-1}_{m,m})^{-1}$ [illegible] $\tilde{F}^1_{n,m} = \tilde{\Phi}^n_{m,m} (U^1_n)^T (\tilde{\Phi}^{n-1}_{m,m})^{-1}$

[illegible]. Equation (3.41) follows by equating the coefficients of $z^n$ in (3.17) on the left. The same argument gives (3.41). To show (3.42) multiply (3.18) on the left by $\Gamma_{n,m}$ and then subtract the resulting equation from (3.16). Now equating the coefficients of $w^m$ gives the result. Equation (3.43) follows by taking the transpose of the reverse of (3.17), then multiplying (3.19) on the left by $\bar{\Gamma}^1_{n,m}$, and subtracting the resulting equations. Equating powers of $w^m$ then gives the result. Equation (3.44) follows by equating the highest powers of $w$ in (3.18), and (3.45) follows in a similar manner from (3.19) and the fact that $C_{n,m}$ is a doubly Toeplitz matrix. □

[illegible] 3.11. From (3.11) and Lemma 2.5 we see that $(\Phi^m_{n,n})^\dagger$ is the lower Cholesky factor of $[I_{m+1}, 0, \dots, 0] C_{n,m} [I_{m+1}, 0, \dots, 0]^T$ and a similar relation holds between $\tilde{\Phi}^n_{m,m}$and $\tilde{C}_{n,m}$. Thus (3.44) and (3.45) give the relation between $I_{n,m}$ and $I^1_{n,m}$ and the Fourier coefficients of $\mathcal{L}_{N,M}$. These coupled with (3.42) and (3.43) relate the Fourier coefficients of $\mathcal{L}_{N,M}$ to $\mathcal{K}_{n,m}$ and $\mathcal{K}^1_{n,m}$.

We now give relations between the coefficients in the recurrence formulas at one level in terms of those at previous levels.

LEMMA 3.12 (relations for $\mathcal{K}_{n,m}$). [illegible] $0 < n, m$

$$\Gamma^1_{n,m-1} \mathcal{K}_{n,m} = \mathcal{K}_{n,m-1} (\tilde{A}^{-1}_{n-1,m})^\dagger - \mathcal{K}^1_{n,m-1} \hat{\bar{E}}^\dagger_{n-1,m} (\tilde{A}^{-1}_{n-1,m})^\dagger, \tag{3.46}$$

$$\mathcal{K}_{n,m} (\tilde{\Gamma}^1_{n-1,m})^\dagger = A^{-1}_{n,m-1} \mathcal{K}_{n-1,m} - A^{-1}_{n,m-1} \hat{E}_{n,m-1} \bar{\mathcal{K}}^1_{n-1,m}. \tag{3.47}$$

[illegible]. To show (3.46) multiply (3.22) on the left by $\Gamma^1_{n,m-1}$ and then use (3.17) with $m$ reduced by one to obtain

$$\Gamma^1_{n,m-1} \mathcal{K}_{n,m} = \langle w \Phi_{n,m-2}, \tilde{\Phi}_{n-1,m} \rangle.$$

Eliminating $\tilde{\Phi}_{n-1,m}$ using ($\tilde{3}$.14) and then applying (3.22) and (3.24) gives (3.46). Equation (3.47) follows in an analogous manner. □

LEMMA 3.13 (relations for $\mathcal{K}^1_{n,m}$). [illegible] $0 < n, m$

$$\Gamma_{n,m-1} \mathcal{K}^1_{n,m} = \mathcal{K}^1_{n,m-1} (\tilde{A}^{-1}_{n-1,m})^T - \mathcal{K}_{n,m-1} (\hat{\bar{E}}_{n-1,m})^T (\tilde{A}^{-1}_{n-1,m})^T, \tag{3.48}$$

$$\mathcal{K}^1_{n,m} (\tilde{\Gamma}_{n-1,m})^T = A^{-1}_{n,m-1} \mathcal{K}^1_{n-1,m} - A^{-1}_{n,m-1} \hat{E}_{n,m-1} \bar{\mathcal{K}}_{n-1,m}. \tag{3.49}$$

[illegible]. To show (3.48) multiply (3.24) on the left by $\Gamma_{n,m-1}$ and then use (3.16) to obtain

$$\Gamma_{n,m-1} \mathcal{K}^1_{n,m} = \langle w \Phi_{n,m-2}, \overleftarrow{\tilde{\Phi}}{}^T_{n-1,m} \rangle.$$

Now use $(\tilde{3}.14)$ with $n$ reduced by one and then (3.24) and (3.22) to find (3.48). Equation (3.49) follows in a similar manner. ☐

LEMMA 3.14 (relations for $\hat{E}_{n,m}$). *Let* $0 < n, m$

$$\Gamma_{n-1,m}\hat{E}_{n,m} = A_{n,m-1}\mathcal{K}_{n,m}(I^1_{n-1,m})^\dagger + \hat{E}_{n,m-1}\bar{\Gamma}^1_{n-1,m}, \tag{3.50}$$

$$\hat{E}_{n,m}(\Gamma^1_{n-1,m})^T = I_{n-1,m}(\mathcal{K}^1_{n,m})^T A^T_{n,m-1} + \Gamma^\dagger_{n-1,m}\hat{E}_{n,m-1}. \tag{3.51}$$

*Proof.* To establish (3.50) multiply (3.20) on the left by $\Gamma_{n-1,m}$ and then use (3.16) to obtain

$$\Gamma_{n-1,m}\hat{E}_{n,m} = \langle z\Phi_{n-1,m-1}, \overleftarrow{\Phi}^T_{n-1,m}\rangle.$$

With the use of (3.14) to eliminate $z\Phi_{n-1,m-1}$, we find

$$\Gamma_{n-1,m}\hat{E}_{n,m} = A_{n,m-1}\langle\Phi_{n,m-1}, \overleftarrow{\Phi}^T_{n-1,m}\rangle + \hat{E}_{n,m-1}\langle\overleftarrow{\Phi}^T_{n-1,m-1}, \overleftarrow{\Phi}^T_{n-1,m}\rangle.$$

The second inner product on the right-hand side of the above equation evaluates to $\bar{\Gamma}^1_{n-1,m}$, while the first may be evaluated using (3.19) followed by (3.22) to give the claimed equation. To obtain (3.51) multiply (3.20) on the right by $(\Gamma^1_{n-1,m})^T$ and then use (3.17) to get

$$\hat{E}_{n,m}(\Gamma^1_{n-1,m})^T = \langle z\Phi_{n-1,m}, \overleftarrow{\Phi}^T_{n-1,m-1}\rangle.$$

Using (3.14) to eliminate $\overleftarrow{\Phi}^T_{n-1,m-1}$ yields

$$\hat{E}_{n,m}(\Gamma^1_{n-1,m})^T = \langle z\Phi_{n-1,m}, \overleftarrow{\Phi}^T_{n,m-1}\rangle A^T_{n,m-1} + \langle\Phi_{n-1,m}, \Phi_{n-1,m-1}\rangle\hat{E}^T_{n,m-1}.$$

Equation (3.23) can be used to evaluate the second inner product on the right-hand side of the above equation, while the reverse transpose of (3.17) and (3.26) can be used to obtain the first inner product. ☐

LEMMA 3.15 (relation for $\Gamma^1_{n,m}$). *Let* $0 < n, m$

$$\begin{aligned}\Gamma^1_{n,m}\Gamma^\dagger_{n,m} = {}& I_{n,m-1}\tilde{\hat{E}}_{n,m}(I^1_{n,m-1})^T + \Gamma^\dagger_{n,m-1}\Gamma^1_{n,m-1}\\ &+ \mathcal{K}^1_{n,m}\tilde{\bar{A}}^{-1}_{n-1,m}\tilde{\hat{E}}^\dagger_{n-1,m}\tilde{A}_{n-1,m}\mathcal{K}^\dagger_{n,m}.\end{aligned} \tag{3.52}$$

*Proof.* To show (3.52) multiply (3.25) on the left by $\Gamma^\dagger_{n,m}$ and use (3.16) to find

$$\Gamma^1_{n,m}\Gamma^\dagger_{n,m} = \langle w\Phi_{n,m-1}, \Phi_{n,m-1}\rangle - \langle w\Phi_{n,m-1}, \tilde{\Phi}_{n-1,m}\rangle\mathcal{K}^\dagger_{n,m}. \tag{3.53}$$

Eliminating $w\Phi_{n,m-1}$ in the second term on the right-hand side of the above equation using (3.17) and then applying $(\tilde{3}.15)$ gives the third term on the right-hand side of (3.52). In the first term on the right-hand side of the above equation, substitute the reverse transpose of (3.19) to find

$$\langle w\Phi_{n,m-1}, \Phi_{n,m-1}\rangle = \langle w\Phi_{n,m-1}, \overleftarrow{\tilde{\Phi}}^T_{n,m-1}\rangle(I^1_{n,m-1})^T + \Gamma^\dagger_{n,m-1}\Gamma^1_{n,m-1},$$

where (3.23) has been used to obtain the second term on the right-hand side of the above equation. The result may now be obtained by applying (3.18) for $\Phi_{n,m-1}$ and then using $(\tilde{3}.20)$. ☐

LEMMA 3.16 (relations for $I_{n,m}$ and $I^1_{n,m}$).

$$I_{n,m}\tilde{\Gamma}^\dagger_{n,m} = -\Gamma^\dagger_{n,m}\mathcal{K}_{n,m}, \tag{3.54}$$

$$I^1_{n,m} = -\bar{A}^{-1}_{n,m}\hat{E}^\dagger_{n,m}A_{n,m}I_{n,m} + A^T_{n,m}I^1_{n-1,m}\tilde{\Gamma}_{n,m}, \quad 0 < n. \tag{3.55}$$

Equation (3.54) follows by multiplying (3.26) on the right by $\tilde{\Gamma}^\dagger_{n,m}$ and then using ($\tilde{3}$.16) and (3.23). In (3.27) use ($\tilde{3}$.18) and (3.28) to find

$$I^1_{n,m} = \langle \overleftarrow{\Phi}^T_{n,m}, \Phi_{n,m}\rangle I_{n,m} + \langle \overleftarrow{\Phi}^T_{n,m}, \tilde{\Phi}_{n-1,m}\rangle \tilde{\Gamma}_{n,m}.$$

The first inner product on the right-hand side may be evaluated using (3.15). To evaluate the second inner product, eliminate $\overleftarrow{\Phi}^T_{n,m}$ using the reverse transpose of (3.15) and then use (3.27) to obtain the claimed equation. □

**4. Christoffel–Darboux formulas.** The Christoffel–Darboux formula plays an important role in the theory of one variable scalar and matrix orthogonal polynomials. Using the connection between two variable orthogonal polynomials and matrix orthogonal polynomials, we derive two variable analogs of the Christoffel–Darboux formula. These will play an important role in the theory of two variable stable polynomials discussed later.

LEMMA 4.1. $\{\Phi_{n,m}\}$ $\{\tilde{\Phi}_{n,m}\}$

$$\begin{aligned}
&\overleftarrow{\Phi}_{n,m}(z,w)\overleftarrow{\Phi}^\dagger_{n,m}(z_1,w_1) - \bar{z}_1 z\Phi^T_{n,m}(z,w)\Phi^\dagger_{n,m}(z_1,w_1)^T && \text{(4.1a)}\\
&\quad = (1-\bar{z}_1 z)\Phi_{n,m}(z,w)^T\Phi^\dagger_{n,m}(z_1,w_1)^T \\
&\quad + \overleftarrow{\Phi}_{n-1,m}(z,w)\overleftarrow{\Phi}^\dagger_{n-1,m}(z_1,w_1) - \bar{z}_1 z\Phi^T_{n-1,m}(z,w)\Phi^\dagger_{n-1,m}(z_1,w_1)^T && \text{(4.1b)}\\
&\quad = (1-\bar{z}_1 z)\tilde{\Phi}_{n,m}(z,w)^T\tilde{\Phi}^\dagger_{n,m}(z_1,w_1)^T \\
&\quad + \overleftarrow{\Phi}_{n,m-1}(z,w)\overleftarrow{\Phi}_{n,m-1}(z_1,w_1)^T - \bar{z}_1 z\Phi_{n,m-1}(z,w)^T\Phi^\dagger_{n,m-1}(z_1,w_1)^T. && \text{(4.1c)}
\end{aligned}$$

The equality (4.1a)=(4.1b) follows by subtracting (2.24) with $n$ reduced by one from the original equation and then using Lemma 3.3. The equality (4.1a)=(4.1c) can be obtained in the following manner. Let

$$Z_{n,m}(z,w) = [1, w, \ldots, w^m][I_{m+1}, zI_{m+1}, \ldots, z^n I_{m+1}],$$

and let $\tilde{Z}_{n,m}(z,w)$ be given by a similar formula with the roles of $z$ and $w$ and $n$ and $m$ interchanged. Then from Lemma 2.5, (2.24), and (3.11) we find

$$\begin{aligned}
&\frac{\overleftarrow{\Phi}_{n,m}(z,w)\overleftarrow{\Phi}^\dagger_{n,m}(z_1,w_1) - \bar{z}_1 z\Phi^T_{n,m}(z,w)\Phi^\dagger_{n,m}(z_1,w_1)^T}{1-\bar{z}_1 z}\\
&\qquad = Z_{n,m}(z,w)C^{-1}_{n,m}Z_{n,m}(z_1,w_1)^\dagger = \tilde{Z}_{n,m}(z,w)\tilde{C}^{-1}_{n,m}\tilde{Z}_{n,m}(z_1,w_1)^\dagger\\
&\qquad = \tilde{\Phi}^T_{n,m}(z,w)\tilde{\Phi}^\dagger_{n,m}(z_1,w_1)^T + \tilde{Z}_{n,m-1}(z,w)\tilde{C}^{-1}_{n,m-1}\tilde{Z}_{n,m-1}(z_1,w_1)^\dagger.
\end{aligned}$$

Switching back to the lexicographical ordering in the second term in the last equation and then using Lemma 2.5 yields the result. □

As an immediate application of the above lemma we obtain the following.

THEOREM 4.2 (Christoffel–Darboux formula). [illegible] $\{\Phi_{n,m}\}$ [illegible] $\{\tilde{\Phi}_{n,m}\}$

$$\begin{aligned}&\frac{\overleftarrow{\Phi}_{n,m}(z,w)\overleftarrow{\Phi}^{\dagger}_{n,m}(z_1,w_1)-\bar{z}_1 z\Phi^T_{n,m}(z,w)\Phi^{\dagger}_{n,m}(z_1,w_1)^T}{1-\bar{z}_1 z}\\ &\qquad=\sum_{k=0}^{n}\Phi^T_{k,m}(z,w)\Phi^{\dagger}_{k,m}(z_1,w_1)^T\\ &\qquad=\sum_{j=0}^{m}\tilde{\Phi}^T_{n,j}(z,w)\tilde{\Phi}^{\dagger}_{n,j}(z_1,w_1)^T.\end{aligned}$$

In the first line of the above equation, the terms $\bar{z}_1 z$ may be replaced by $\bar{w}_1 w$ if we switch to $\tilde{\Phi}_{n,m}$.

An interesting variant of (4.1c) is the following.

LEMMA 4.3.

$$\begin{aligned}&\Phi_{n,m}(z,w)^T\Phi^{\dagger}_{n,m}(z_1,w_1)^T-\Phi^T_{n,m-1}(z,w)\Phi^{\dagger}_{n,m-1}(z_1,w_1)^T\\ &=\ \tilde{\Phi}_{n,m}(z,w)^T\tilde{\Phi}^{\dagger}_{n,m}(z_1,w_1)^T-\tilde{\Phi}^T_{n-1,m}(z,w)\tilde{\Phi}^{\dagger}_{n-1,m}(z_1,w_1)^T.\end{aligned}\tag{4.2}$$

[illegible]. Equating the sums in the above theorem yields

$$\Phi_{n,m}(z,w)^T\Phi^{\dagger}_{n,m}(z_1,w_1)^T-\sum_{j=0}^{m-1}\tilde{\Phi}^T_{n,j}(z,w)\tilde{\Phi}^{\dagger}_{n,j}(z_1,w_1)^T\tag{4.3}$$

$$=\tilde{\Phi}_{n,m}(z,w)^T\tilde{\Phi}^{\dagger}_{n,m}(z_1,w_1)^T-\sum_{j=0}^{n-1}\Phi^T_{j,m}(z,w)\Phi^{\dagger}_{j,m}(z_1,w_1)^T.\tag{4.4}$$

Switching to the lexicographical ordering in the sum on the left-hand side of the above equation and reverse lexicographical ordering in the sum on the right-hand side, extracting the highest terms, and then using the Christoffel–Darboux formula to eliminate the remaining sums gives the result. ☐

[illegible] 4.4. The above equations can be derived from the recurrence formulas in the previous sections. However, the derivation of (4.1c) is rather tedious.

**5. Algorithm.** In this section we use the relations developed earlier to provide an algorithm that allows us to compute the coefficients in the recurrence formula at higher levels in terms of those at lower levels plus some indeterminates that are equivalent to the moments. This will allow us to construct positive definite doubly Toeplitz matrices. As a byproduct we construct the orthogonal polynomials associated with these matrices. More precisely, at each level we use the new indeterminates and the coefficients on the levels $(n, m-1)$ and $(n-1, m)$ to construct $\mathcal{K}_{n,m}$ and $\mathcal{K}^1_{n,m}$. With this we can construct the other coefficients needed to proceed to the next level. The $\hat{E}_{n,m}$ are closely related to the matrix recurrence coefficients needed to compute $C_{n,m}$. Furthermore $\Phi_{n,m}$ and $\tilde{\Phi}_{n,m}$ can also be computed. In order to construct the above matrices we will have need of the $m\times(m+1)$ matrices $U_m$ and $U^1_m$ given by (3.35) and (3.36), respectively, and the vector $e^m_1\in\mathbb{R}^m$, which is the vector with one in the first entry and zeros everywhere else. From the definition of $\mathcal{K}^1_{n,m}$ we see that

$$\begin{aligned}\mathcal{K}^1_{n,m} &= \langle w\Phi_{n,m-1}, \overleftarrow{\tilde{\Phi}}^T_{n-1,m}\rangle \\ &= \left\langle w\begin{bmatrix}\phi^{m-1}_{n,m-1}\\ \vdots \\ \phi^0_{n,m-1}\end{bmatrix}, \begin{bmatrix}\overleftarrow{\tilde{\phi}}^{n-1}_{n-1,m}\\ \vdots\\ z^n\overleftarrow{\tilde{\phi}}^0_{n-1,m}\end{bmatrix}\right\rangle \\ &= c_{-n,-m}d_{n,m}e_1^m(e_1^n)^T + R_{n,m}, \qquad (5.1)\end{aligned}$$

where $R_{n,m}$ is an $m\times n$ matrix containing moments $c_{i,j}, \{|i| \le n, |j| \le m\}\backslash\{(\pm n, \pm m)\}$. Likewise, with the help of (3.38) and its tilde counterpart we find

$$\begin{aligned}\mathcal{K}_{n,m} &= \langle \Phi_{n,m-1}, \tilde{\Phi}_{n-1,m}\rangle \\ &= c_{-n,m}\Phi^{n,m-1}_{n,m-1}\begin{bmatrix}0\\ \vdots\\ 1\end{bmatrix}\left(\tilde{\Phi}^{m,n-1}_{m,n-1}\begin{bmatrix}0\\ \vdots\\ 1\end{bmatrix}\right)^{\dagger} + \hat{R}_{n,m}, \qquad (5.2)\end{aligned}$$

where $\hat{R}_{n,m}$ contains only moments from lower levels.

We proceed as follows: At level $(0,0)$ we have the parameter $u_{0,0} > 0$, which corresponds to $c_{0,0}$. The polynomials $\Phi_{0,0}$ and $\tilde{\Phi}_{0,0}$ are chosen as $\frac{1}{\sqrt{u_{0,0}}}$. From (3.26) and (3.27) we see that $I_{0,0} = 1 = I^1_{0,0}$. At level $(i,0)$ there is one new parameter $u_{i,0}$, which can be taken to correspond with the one-dimensional recurrence coefficient i.e., $u_{i,0} = \alpha_i = \hat{E}_{i,0}$, corresponding to the $(i,0)$ level and must be less than one in magnitude. From (3.30) and the normalization chosen for the polynomials, $A_{i,0} = \sqrt{1-|\hat{E}_{i,0}|^2}$. This allows us to compute $\Phi_{i,0}$, and $\overleftarrow{\Phi}_{i,0}$. The sizes of the matrices given in (3.22), (3.23), (3.24), and (3.25) show that

$$\mathcal{K}_{i,0} = \tilde{\mathcal{K}}_{i,0} = \Gamma_{i,0} = \mathcal{K}^1_{i,0} = \tilde{\mathcal{K}}^1_{i,0} = \Gamma^1_{i,0} = 0,$$

where (3.28) and (3.29) have also been used. Furthermore (3.33) and Remark 3.9 imply that $I_{i,0} = (e_1^{i+1})^T = \tilde{I}^{\dagger}_{i,0}$ where (3.28) has been used. Equation $(\tilde{3}.31)$ implies that $\tilde{\Gamma}_{i,0} = U_i$, while (3.55) and (3.29) allow us to compute

$$I^1_{i,0} = (\tilde{I}^1_{i,0})^T = -\begin{bmatrix}\hat{E}^{\dagger}_{i,0} & A_{i,0}\hat{E}^{\dagger}_{i-1,0} & \cdots & -\prod_{j=1}^{i} A_{j,0}\end{bmatrix}. \qquad (5.3)$$

$\tilde{\Phi}_{i,0}$ can now be computed from $(\tilde{3}.18)$.

At level $(0,j)$ there is one new parameter $u_{0,j}$, which as above can be taken to correspond with the one-dimensional recurrence coefficient, i.e., $u_{0,j} = \alpha_j = \tilde{\hat{E}}_{0,j}$, corresponding to the $(0,j)$ level and must be less than one in magnitude. The analysis for the $(i,0)$ level can be carried over with the roles of the lexicographical and reverse lexicographical orderings interchanged. Thus from $(\tilde{3}.30)$ and the normalization chosen for the polynomials, $\tilde{A}_{0,j} = \sqrt{1-|\tilde{\hat{E}}_{0,j}|^2}$, which allows us to compute $\tilde{\Phi}_{0,j}$. Again

$$\tilde{\mathcal{K}}_{0,j} = \mathcal{K}_{0,j} = \tilde{\Gamma}_{0,j} = \tilde{\mathcal{K}}^1_{0,j} = \mathcal{K}^1_{0,j} = \tilde{\Gamma}^1_{0,j} = 0.$$

Likewise $\tilde{I}_{0,j} = (e_1^{j+1})^T = I^{\dagger}_{0,j}$ and $\Gamma_{0,j} = U_j$. Equations $(\tilde{3}.55)$ and (3.29) allow us to compute $\tilde{I}^1_{0,j}$ as above with $i$ and $j$ interchanged as well as the orderings. Equation (3.18) now allows us to compute $\Phi_{0,j}$.

At level $(n,m)$, with $n,m>0$, there are two new parameters $u_{n,m}$ and $u_{-n,m}$ since $u_{-n,-m}=\overline{u}_{n,m}$ and $u_{n,-m}=\overline{u}_{-n,m}$. These along with the coefficients on the $(n-1,m)$ and $(n,m-1)$ level will be used to compute $\mathcal{K}_{n,m}$ and $\mathcal{K}^1_{n,m}$. This will be sufficient to compute the remaining coefficients on level $(n,m)$. We begin with the following.

$\mathcal{K}_{n,m}$ If $n=1, m=1$, then (3.22) shows that $\mathcal{K}_{1,1}$ is a scalar which we choose as $\bar{u}_{1,-1}$. If $m>1$, we see from (3.41) and (3.36) that

$$\Gamma^1_{n,m-1}\Phi^{m-1}_{n,n}e^m_m=0,$$

where $e^m_m$ is the $m$-dimensional vector with zeros in all its entries except the last, which is one. Since $\Phi^{m-1}_{n,n}=A^{-1}_{n,m-1}\cdots A^{-1}_{1,m-1}\Phi^{m-1}_{0,0}$ is an upper triangular invertible matrix, we find

$$\Gamma^1_{n,m-1}\Phi^{m-1}_{n,n}((U^1_{m-1})^TU^1_{m-1}+e^m_m(e^m_m)^T)=\Gamma^1_{n,m-1}\Phi^{m-1}_{n,n}(U^1_{m-1})^TU^1_{m-1},$$

and from (3.41) $\Gamma^1_{n,m-1}\Phi^{m-1}_{n,n}(U^1_{m-1})^T=\Phi^{m-2}_{n,n}$. Thus (3.46) can be written as

$$\begin{aligned}
&U^1_{m-1}(\Phi^{m-1}_{n,n})^{-1}\mathcal{K}_{n.m}((\tilde{\Phi}^{n-1}_{m,m})^\dagger)^{-1}\\
&=(\Phi^{m-2}_{n,n})^{-1}(\mathcal{K}_{n,m-1}(\tilde{A}^{-1}_{n-1,m})^\dagger-\mathcal{K}^1_{n,m-1}\hat{\tilde{E}}^\dagger_{n-1,m}(\tilde{A}^{-1}_{n-1,m})^\dagger)((\tilde{\Phi}^{n-1}_{m,m})^\dagger)^{-1}\\
&=(\Phi^{m-2}_{n,n})^{-1}(\mathcal{K}_{n,m-1}-\mathcal{K}^1_{n,m-1}\hat{\tilde{E}}^\dagger_{n-1,m})((\tilde{\Phi}^{n-1}_{m-1,m-1})^\dagger)^{-1}\\
&=H_{n,m-1}.
\end{aligned}$$

In the last equality we have used the fact that $\tilde{A}_{n-1,m}\tilde{\Phi}^{n-1}_{m,m}=\tilde{\Phi}^{n-1}_{m-1,m-1}$. Likewise,

$$\begin{aligned}
&(\Phi^{m-1}_{n,n})^{-1}\mathcal{K}_{n,m}((\tilde{\Phi}^{n-1}_{m,m})^{-1})^\dagger(U^1_{n-1})^T\\
&=(\Phi^{m-1}_{n-1,n-1})^{-1}(\mathcal{K}_{n-1,m}-\hat{E}_{n,m-1}\bar{\mathcal{K}}^1_{n-1,m})((\tilde{\Phi}^{n-2}_{m,m})^\dagger)^{-1}\\
&=\tilde{H}_{n-1,m}.
\end{aligned}$$

If

$$(e^m_m)^T(\Phi^{m-1}_{n,n})^{-1}\mathcal{K}_{n,m}((\tilde{\Phi}^{n-1}_{m,m})^\dagger)^{-1}e^n_n=\bar{u}_{n,-m}, \tag{5.4}$$

then $\mathcal{K}_{n,m}$ can be solved for as

(5.5)

$$\begin{aligned}
&\mathcal{K}_{n,m}\\
&=\Phi^{m-1}_{n,n}\left(u_{-n,m}e^m_m(e^n_n)^T+(U^1_{m-1})^TH_{n,m-1}+e^m_m(e^m_m)^T\tilde{H}_{n-1,m}(U^1_{n-1})\right)(\tilde{\Phi}^{n-1}_{m,m})^\dagger.
\end{aligned}$$

A necessary condition in order to be able to continue is that $||\mathcal{K}_{n,m}||<1$.

$\Gamma_{n,m}$ Since $\mathcal{K}_{n,m}$ is presumed to be a contraction, Remark 3.9 shows that $\Gamma_{n,m}$ and $\tilde{\Gamma}_{n,m}$ may be computed from the upper Cholesky factor of $I-\mathcal{K}_{n,m}\mathcal{K}^\dagger_{n,m}$ and $I-\mathcal{K}^\dagger_{n,m}\mathcal{K}_{n,m}$, respectively.

$\mathcal{K}^1_{n,m}$ In $\mathcal{K}^1_{n,m}$ we see from (5.1) that the only new entry is $(\mathcal{K}_{n,m})_{1,1}$. If $n=1, m=1$, set $\mathcal{K}_{1,1}=\bar{u}_{1,1}$. If $m>1$, we will show that all of the rows except the first can be obtained from (3.48). The structure of $\Gamma_{n,m-1}$ implies that $\Gamma_{n,m-1}e^m_1=0$ so that

$$\Gamma_{n,m-1}=\Gamma_{n,m-1}(U^T_{m-1}U_{m-1}+e^m_1(e^m_1)^T)=\Gamma_{n,m-1}U^T_{m-1}U_{m-1}.$$

But $\Gamma_{n,m-1}U_{m-1}^T$ is an invertible matrix, which allows us to rewrite (3.48) as follows:

$$U_{m-1}\mathcal{K}_{n,m}^1 = (\Gamma_{n,m-1}U_{m-1}^T)^{-1}(\mathcal{K}_{n,m-1}^1(\tilde{A}_{n-1,m}^{-1})^T - \mathcal{K}_{n,m-1}\hat{\hat{E}}_{n-1,m}^T(\tilde{A}_{n-1,m}^{-1})^T).$$

This gives all of the entries in $\mathcal{K}_{n,m}^1$ except the first row.

Similarly, if $n > 1$ we can write

$$\tilde{\Gamma}_{n-1,m} = \tilde{\Gamma}_{n-1,m}(U_{n-1}^T U_{n-1} + e_1^n(e_1^n)^T) = \tilde{\Gamma}_{n-1,m}U_{n-1}^T U_{n-1},$$

i.e., $\tilde{\Gamma}_{n-1,m}^T = U_{n-1}^T U_{n-1}\tilde{\Gamma}_{n-1,m}^T$, and (3.49) can be rewritten as

$$\mathcal{K}_{n,m}^1 U_{n-1}^T = (A_{n,m-1}^{-1}\mathcal{K}_{n-1,m}^1 - A_{n,m-1}^{-1}\hat{E}_{n,m-1}\bar{\mathcal{K}}_{n-1,m})(U_{n-1}\tilde{\Gamma}_{n-1,m}^T)^{-1}. \tag{5.6}$$

Thus the $m \times (n-1)$ matrix $\mathcal{K}_{n,m}^1 U_{n-1}^T$, which is obtained from $\mathcal{K}_{n,m}^1$ by deleting the first column, is known from the previous levels. This allows us to compute all entries in the first row of $\mathcal{K}_{n,m}^1$ except $(\mathcal{K}_{n,m}^1)_{1,1}$, and we put

$$(\mathcal{K}_{n,m}^1)_{1,1} = \bar{u}_{n,m}. \tag{5.7}$$

A necessary condition on the parameters in order to be able to continue is that $||\mathcal{K}_{n,m}^1|| < 1$, which implies that $|u_{n,m}| < 1$.

[illegible] $\hat{E}_{n,m}$. We begin by taking the transpose of (3.51), using the fact that $\hat{E}_{n,m}$ is symmetric, and then multiplying on the left by the matrix $e_1^{m+1}(e_1^m)^T$. Now multiply (3.50) by $U_m^T$ and add the resulting equations. If $\hat{\Gamma}_{n-1,m}$ is the $(m+1) \times (m+1)$ matrix obtained by stacking the first row of $\Gamma_{n-1,m}^1$ on $\Gamma_{n-1,m}$, we find

$$\begin{aligned}\hat{\Gamma}_{n-1,m}\hat{E}_{n,m} =& U_m^T(A_{n,m-1}\mathcal{K}_{n,m}(I_{n-1,m}^1)^\dagger + \hat{E}_{n,m-1}\bar{\Gamma}_{n-1,m}^1)\\ &+ e_1^{m+1}(e_1^m)^T(A_{n,m-1}\mathcal{K}_{n,m}^1 I_{n-1,m}^T + \hat{E}_{n,m-1}\bar{\Gamma}_{n-1,m}).\end{aligned} \tag{5.8}$$

From the structure of $\Gamma^1$ and $\Gamma$ we see that $\hat{\Gamma}_{n-1,m}$ is an upper triangular matrix with positive diagonal entries and is hence invertible. Thus $\hat{E}_{n,m}$ can be computed from the above equation. If $||\hat{E}_{n,m}|| < 1$, then $\hat{\hat{E}}_{n,m}$ may be computed from ($\tilde{3}.50$) and ($\tilde{3}.51$). We may also compute $A_{n,m}$, $\tilde{A}_{n,m}$, and the polynomials $\Phi_{n,m}$ and $\tilde{\Phi}_{n,m}$. While the condition that $\hat{E}_{n,m}$ be a contraction is necessary and sufficient to be able to continue, it is not optimal in the sense that it does not take into account the redundancy inherent in the equations giving $\hat{E}_{n,m}$. This will be taken into account in the computation of $\Gamma_{n,m}^1$.

[illegible] $\Gamma_{n,m}^1$. As above we see that (3.52) gives

$$\begin{aligned}\Gamma_{n,m}^1 U_m^T =& (I_{n,m-1}\hat{\hat{E}}_{n,m}(I_{n,m-1}^1)^T + \Gamma_{n,m-1}^\dagger\Gamma_{n,m-1}^1\\ &+ \mathcal{K}_{n,m}^1\bar{\bar{\tilde{A}}}_{n-1,m}^{-1}\hat{\hat{E}}_{n-1,m}^\dagger\tilde{A}_{n-1,m}\mathcal{K}_{n,m}^\dagger)(U_m\Gamma_{n,m}^\dagger)^{-1},\end{aligned} \tag{5.9}$$

which allows the computation of all of the entries of $\Gamma_{n,m}^1$ except the $(1,1)$ entry. Since $(e_1^m)^T I_{n,m-1} = (e_1^m)^T$, $(e_1^m)^T\Gamma_{n,m-1}^\dagger = 0$, and likewise, with $I_{n,m-1}$ and $\Gamma_{n,m-1}^\dagger$ replaced by $\tilde{I}_{n,m}$ and $\tilde{\Gamma}_{n,m-1}^T$, respectively, we find with the help of (5.8)

$$(e_1^m)^T\Gamma_{n,m}^1 U_m^T = (e_1^m)^T H_{n,m}^2 + (e_1^m)^T\mathcal{K}_{n,m}^1 H_{n,m}^1, \tag{5.10}$$

where

$$H^2_{n,m} = I_{n,m-1}((I^1_{n,m-1})^\dagger \bar{\mathcal{K}}_{n,m}\tilde{A}^T_{n-1,m} + (\tilde{\Gamma}^1_{n,m-1})^\dagger \hat{\tilde{E}}_{n-1,m}) \\ \times U_n(\tilde{\tilde{\Gamma}}^T_{n,m-1})^{-1}(I^1_{n,m-1})^T(U_m\Gamma^\dagger_{n,m})^{-1}, \tag{5.11}$$

and

$$H^1_{n,m} = \tilde{A}^T_{n-1,m}e^n_1(e^{n+1}_1)^T(\tilde{\tilde{\Gamma}}^T_{n,m-1})^{-1}(I^1_{n,m-1})^T(U_m\Gamma^\dagger_{n,m})^{-1} \\ + \bar{\tilde{A}}^{-1}_{n-1,m}\hat{\tilde{E}}^\dagger_{n-1,m}\tilde{A}_{n-1,m}\mathcal{K}^\dagger_{n,m}(U_m\Gamma^\dagger_{n,m})^{-1}. \tag{5.12}$$

Thus the first entry $\Gamma^1$ can be computed using the first row of (5.10) and (3.32), which gives

$$|(\Gamma^1_{n,m})_{(1,1)}|^2 = 1 - (e^m_1)^T H^3_{n,m}(e^m_1), \tag{5.13}$$

where

$$H^3_{n,m} = (H^2_{n,m} + \mathcal{K}^1_{n,m}H^1_{n,m})(H^2_{n,m} + \mathcal{K}^1_{n,m}H^1_{n,m})^\dagger + \mathcal{K}^1_{n,m}(\mathcal{K}^1_{n,m})^\dagger. \tag{5.14}$$

[illegible]. Using the arguments above we see that the relevant part of $I_{n,m}$ may be computed from (3.54) and $I^1_{n,m}$ may be computed from (3.55). The matrix $\tilde{\Gamma}^1_{n,m}$ can be computed in the same manner as $\Gamma^1_{n,m}$.

## 6. Construction of a positive linear functional.

The above algorithm allows us to find a linear functional given the coefficients in the recurrence formulas. More precisely, it is as follows.

THEOREM 6.1. [illegible] $u_{i,j} \in \mathbb{C}$ $0 \le i \le n, |j| \le m$ $u_{-i,j} = \bar{u}_{i,-j}$ [illegible]

- [illegible] $\hat{E}_{i,0}$ $i = 1, \dots, n$ [illegible] $\hat{\tilde{E}}_{0,j}$ $j = 1, \dots, m$.
- $i \times j$ [illegible] $\mathcal{K}_{i,j}$ $i = 1, \dots, n$ $j = 1, \dots, m$. [illegible]
- $i \times j$ [illegible] $(e^j_1)^T H^3_{i,j}e^j_1$ $i = 1, \dots, n$ $j = 1, \dots, m$

[illegible]

$$u_{0,0} > 0,\ |\hat{E}_{i,0}| < 1, |\hat{\tilde{E}}_{0,j}| < 1,\ ||\mathcal{K}_{i,j}|| < 1, \text{ [illegible] } {e^j_1}^T H^3_{i,j}e^j_1 < 1, \tag{6.1}$$

[illegible] $\mathcal{L}$ [illegible] $\prod^{n,m}$ [illegible]

$$\mathcal{L}(\Phi_{i,m}\Phi^\dagger_{j,m}) = \delta_{i,j}I_{m+1} \text{ [illegible] } \mathcal{L}(\tilde{\Phi}_{n,i}\tilde{\Phi}^\dagger_{n,j}) = \delta_{i,j}I_{n+1}. \tag{6.2}$$

[illegible] (6.1) [illegible]

[illegible]. We construct the linear functional by induction. First, if $n = m = 0$ we set

$$\mathcal{L}(1) = u_{0,0} \text{ and } \Phi_{0,0} = \tilde{\Phi}_{0,0} = \frac{1}{\sqrt{u_{0,0}}},$$

and thus $\mathcal{L}(\Phi_{0,0}\Phi^\dagger_{0,0}) = \mathcal{L}(\tilde{\Phi}_{0,0}\tilde{\Phi}^\dagger_{0,0}) = 1$.

If $m = 0$, we construct $A_{i,0} = \sqrt{1 - |\hat{E}_{i,0}|^2}$, where $\hat{E}_{i,0} = u_{i,0}$. The polynomials $\Phi_{i,0}, i = 0, \dots n$, are now computed using (3.14), and then we define

$$\mathcal{L}(\Phi_{i,0}\Phi^\dagger_{j,0}) = \delta_{i,j}.$$

This gives a well-defined positive linear functional on $z^j$ for $|j| \le n$.

Likewise, if $n = 0$, we construct $\tilde{\Phi}_{0,k}$ using ($\tilde{3}$.14) and define

$$\mathcal{L}(\tilde{\Phi}_{0,i}\tilde{\Phi}_{0,j}^{\dagger}) = \delta_{i,j},$$

which gives the linear functional on $w^j$ for $|j| \leq m$. Thus (6.2) will hold if $m = 0$ or $n = 0$.

Assume now that the functional $\mathcal{L}$ is well defined and positive for all levels $0 \leq i \leq n-1$, $0 \leq j \leq m$ and $0 \leq i \leq n$, $0 \leq j \leq m-1$ before $(n, m)$. To ease notation we will use the bracket given in (3.8) with $\mathcal{L}_{N,M}$ replaced by $\mathcal{L}$. We first extend $\mathcal{L}$ so that

$$\langle \Phi_{n,m-1}, \tilde{\Phi}_{n-1,m} \rangle = \mathcal{K}_{n,m}. \tag{6.3}$$

To check that the above equation is consistent with how $\mathcal{L}$ is defined on the previous levels, note that from (3.46)

$$\langle \Gamma^1_{n,m-1}\Phi_{n,m-1}, \tilde{\Phi}_{n-1,m} \rangle = \Gamma^1_{n,m-1}\mathcal{K}_{n,m}, \tag{6.4}$$

which follows from the construction of $\mathcal{K}_{n,m}$ and the definition of $\mathcal{L}$ on the previous levels (see Lemma 3.12). Similarly, using the second defining relation of $\mathcal{K}_{n,m}$ (i.e., the last row of (3.47)) we see that

$$\langle \Phi_{n,m-1}, \tilde{\Gamma}^1_{n-1,m}\tilde{\Phi}_{n-1,m} \rangle = \mathcal{K}_{n,m}(\tilde{\Gamma}^1_{n-1,m})^{\dagger}. \tag{6.5}$$

Equations (6.4) and (6.5) show that most of (6.3) is automatically true. We now define $\mathcal{L}(z^n w^{-m})$ so that (5.4) holds, which completes (6.3).

Using an analogous argument we can use the construction of $\mathcal{K}^1_{n,m}$ to extend the functional to $z^n w^m$ so that

$$\mathcal{K}^1_{n,m} = \langle w\Phi_{n,m-1}, \overleftarrow{\tilde{\Phi}}{}^{T}_{n-1,m} \rangle. \tag{6.6}$$

This completes the extension of $\mathcal{L}$. What remains to be shown is that (6.2) holds. This is accomplished by first constructing $\tilde{\tilde{E}}_{n,m}$ from ($\tilde{5}$.8). The condition on $(e_1^m)^T H^3_{n,m} e_1^m$ and (5.9) shows that the first row of $\Gamma^1_{n,m}$ may be computed and that we may choose

$$(\Gamma^1_{n,m})_{1,1} > 0.$$

With the first row of $\Gamma^1_{n,m}$ and all of $\Gamma_{n,m}$ (which is calculated from the Cholesky factorization of $\mathcal{K}_{n,m}\mathcal{K}^{\dagger}_{n,m}$), $\Phi_{n,m}$ may be constructed from (3.16) and (3.17). Equations (6.3) and (6.6), coupled with (3.16), (3.17), and the orthogonality relations on the previous levels show that

$$\langle \Gamma_{nm}\Phi_{n,m},\ \tilde{\Phi}_{n-1,k} \rangle = 0, \quad k = 0, 1, \ldots, m,$$

and

$$\left\langle (e_1^m)^T \Gamma^1_{nm}\Phi_{n,m},\ w^k \begin{bmatrix} z^{n-1} \\ \vdots \\ 1 \end{bmatrix} \right\rangle = 0, \quad k = 1, 2, \ldots, m. \tag{6.7}$$

Equations (6.7) and (3.17) show

$$0 = \langle (e_1^m)^T \Gamma^1_{n,m}\Phi_{n,m},\ \overleftarrow{\tilde{\Phi}}{}^{T}_{n-1,m} \rangle = \left\langle (e_1^m)^T \Gamma^1_{nm}\Phi_{n,m},\ \begin{bmatrix} z^{n-1} \\ \vdots \\ 1 \end{bmatrix} \right\rangle.$$

The fact that $\overleftarrow{\tilde{\Phi}}^{T}_{n-1,m}$ has an invertible coefficient multiplying

$$\begin{bmatrix} z^{n-1} \\ \vdots \\ 1 \end{bmatrix}$$

has been used to obtain the second equality in the above equation. The above implies that

$$\langle \Phi_{n,m},\ \tilde{\Phi}_{n-1,k} \rangle = 0, \qquad k = 0, 1, \ldots, m,$$

which in turn implies that

$$\langle \Phi_{n,m},\ \Phi_{j,m} \rangle = 0, \qquad j = 0, 1, \ldots, n-1.$$

To show that

$$\langle \Phi_{n,m},\ \Phi_{n,m} \rangle = I_{m+1}$$

we note that (3.16), (3.17), and (3.52) imply that

$$\langle (e_1^m)^T \Gamma^1_{nm} \Phi_{n,m},\ \Gamma_{nm} \Phi_{n,m} \rangle = (e_1^m)^T \Gamma^1_{n,m} \Gamma^\dagger_{n,m}$$

and (3.32) implies that

$$\langle (e_1^m)^T \Gamma^1_{nm} \Phi_{n,m},\ (e_1^m)^T \Gamma^1_{nm} \Phi_{nm} \rangle = (e_1^m)^T \Gamma^1_{nm} (\Gamma^1_{nm})^\dagger (e_1^m).$$

Thus $\mathcal{L}$ is a positive linear functional. The orthogonality relations for the polynomials $\tilde{\Phi}_{i,j}$ now follow. □

Let $C(\mathbb{T}^2)$ denote the set of continuous functions on the bicircle; above the theorem now allows the following.

THEOREM 6.2. [illegible] $u_{i,j} \in \mathbb{C}$ [illegible] $u_{i,-j} = \bar{u}_{-i,j}$ [illegible] (6.1) [illegible] $0 \le i, j,$ [illegible] $\mu$ [illegible] $f \in C(\mathbb{T}^2)$

$$\mathcal{L}(f) = \left(\frac{1}{2\pi}\right)^2 \int_{\mathbb{T}^2} f(\theta, \phi) d\mu(\theta, \phi).$$

[illegible] From the hypotheses imposed above, Theorem 6.1 shows that $C_{n,m}$ is positive definite for all $n$ and $m$, so the result follows from Bochner's theorem [18, section 1.4.3]. □

[illegible] 6.3. The above construction gives a criterion for the existence of a one step extension of the functional. That is, given moments so that there exists a positive linear functional on $\prod^{n-1,m} \cup \prod^{n,m-1}$, any set

$$\{u_{n,m},\ u_{-n,m}\},\ u_{-n,-m} = \bar{u}_{n,m}, \quad u_{n,-m} = \bar{u}_{-n,m},$$

that satisfies (6.1) can be used to extend the functional to $\prod^{n,m}$. However it is not difficult to construct examples where no extension exists. See section 8.

**7. Two variable stable polynomials and Fejér–Riesz factorization.** In this section we study the consequences of $\mathcal{K}_{n,m} = 0$. This will make a connection with the results in [8] on stable polynomials and the Fejér–Riesz factorization theorem.

We say that a polynomial $p(z,w)$ is stable if $p(z,w)\neq 0,\ |z|\leq 1,\ |w|\leq 1$. A polynomial $p$ is of degree $(n,m)$ if

$$p(z,w)=\sum_{i=0}^{n}\sum_{j=0}^{m}k_{i,j}z^iw^j,$$

with $k_{n,m}\neq 0$. Finally we say that the polynomial $p_{n,m}$ of degree $(n,m)$ has the spectral matching property (up to $(n,m)$) if

$$\mathcal{L}(z^kw^j)=\frac{1}{(2\pi)^2}\int_{\mathbb{T}^2}\frac{z^kw^j}{|p_{n,m}(z,w)|^2}d\theta d\phi,\quad z=e^{i\theta},\ w=e^{i\phi},$$

for $|k|\leq n,\ |j|\leq m$.

LEMMA 7.1. [illegible] $\mathcal{L}$ [illegible] $\prod^{n,m}$ [illegible] $\mathcal{K}_{n,m}=0$. [illegible]

$$\begin{aligned}&\overleftarrow{\phi}^{m}_{n,m}(z,w)\overline{\overleftarrow{\phi}^{m}_{n,m}(z_1,w_1)}-\phi^{m}_{n,m}(z,w)\overline{\phi^{m}_{n,m}(z_1,w_1)}\\&\quad=(1-w\bar{w}_1)\overleftarrow{\Phi}_{n,m-1}(z,w)\overleftarrow{\Phi}^{\dagger}_{n,m-1}(z_1,w_1)\\&\qquad+(1-z\bar{z}_1)\tilde{\Phi}_{n-1,m}(z,w)^T\tilde{\Phi}^{\dagger}_{n-1,m}(z_1,w_1).\end{aligned}\tag{7.1}$$

[illegible]. If $\mathcal{K}_{n,m}=0$, then (3.16) shows that $\Gamma_{n,m}\Phi_{n,m}(z,w)=\Phi_{n,m-1}(z,w)$. Thus $\overleftarrow{\Phi}_{n,m}(z,w)\Gamma^{\dagger}_{n,m}=w\overleftarrow{\Phi}_{n,m-1}(z,w)$. Also (3.31) implies that $(\Gamma_{n,m})_{(i,i+1)}=1,\ i=1,\dots,m$ with all other entries zero. Thus we find

$$\begin{aligned}&\overleftarrow{\Phi}_{n,m}(z,w)\overleftarrow{\Phi}_{n,m}(z_1,w_1)^{\dagger}\\&\quad=\overleftarrow{\phi}^{m}_{n,m}(z,w)\overline{\overleftarrow{\phi}^{m}_{n,m}(z_1,w_1)}+\overleftarrow{\Phi}_{n,m}(z,w)\Gamma^{\dagger}_{n,m}\Gamma_{n,m}\overleftarrow{\Phi}_{n,m}(z_1,w_1)^{\dagger}\\&\quad=\overleftarrow{\phi}^{m}_{n,m}(z,w)\overline{\overleftarrow{\phi}^{m}_{n,m}(z_1,w_1)}+w\bar{w}_1\overleftarrow{\Phi}_{n,m-1}(z,w)\overleftarrow{\Phi}_{n,m-1}(z_1,w_1)^{\dagger}.\end{aligned}\tag{7.2}$$

From (4.1c) observe that

$$\begin{aligned}&\overleftarrow{\phi}^{m}_{n,m}(z,w)\overline{\overleftarrow{\phi}^{m}_{n,m}(z_1,w_1)}-z\bar{z}_1\phi^{m}_{n,m}(z,w)\overline{\phi^{m}_{n,m}(z_1,w_1)}\\&\quad=(1-w\bar{w}_1)\overleftarrow{\Phi}_{n,m-1}(z,w)\overleftarrow{\Phi}^{\dagger}_{n,m-1}(z_1,w_1)\\&\qquad+(1-z\bar{z}_1)\tilde{\Phi}_{n,m}(z,w)^T\tilde{\Phi}^{\dagger}_{n,m}(z_1,w_1).\end{aligned}$$

Using ($\tilde{3}$.16) and the fact that $\tilde{\phi}^{n}_{n,m}(z,w)=\phi^{m}_{n,m}$ gives the result. □

We now have the following.

THEOREM 7.2. [illegible] $\mathcal{L}$ [illegible] $\prod^{n,m}$ [illegible] $\mathcal{K}_{n,m}=0$. [illegible] $\overleftarrow{\phi}^{m}_{n,m}(z,w)$ [illegible]

$$\mathcal{L}(e^{-ik\theta}e^{-il\phi})=\left(\frac{1}{2\pi}\right)^2\int_{\mathbb{T}^2}\frac{e^{-ik\theta}e^{-il\phi}}{|\phi^{m}_{n,m}(e^{i\theta},e^{i\phi})|^2}d\theta d\phi,\quad |k|\leq n,\ |l|\leq m.$$

[illegible] $\pi_{n,m}(z,w)$ [illegible] $(n,m)$ [illegible] $\overleftarrow{\pi}_{n,m}$ [illegible]

$$\mathcal{L}(e^{-ik\theta}e^{-il\phi})=\left(\frac{1}{2\pi}\right)^2\int_{\mathbb{T}^2}\frac{e^{-ik\theta}e^{-il\phi}}{|\pi_{n,m}(e^{i\theta},e^{i\phi})|^2}d\theta d\phi,\quad |k|\leq n,\ |l|\leq m,$$

[illegible] $\mathcal{K}_{n,m}=0$

[illegible]. If $\mathcal{L}$ is positive definite and $\mathcal{K}_{n,m}$ is equal to zero, then Lemma 7.1 shows that $\overleftarrow{\phi}^{\,n}_{n,m}(z,w)$ satisfies (7.1). The first part of the result now follows from the proof of Theorem 2.3.1 in [8]. To show the second part, let $f(z,w)=1/|\overleftarrow{\pi}_{n,m}(z,w)|^2, |z|=1=|w|$ be the spectral density function associated with $\pi_{n,m}$. Then from (2.1.5) in [8] and Lemma 3.3 we find that

$$\overleftarrow{\Phi}_{n,m}(z,w)=[\overleftarrow{\pi}_{n,m}(z,w), w\overleftarrow{\Phi}_{n,m-1}(z,w)].$$

But this implies that $\Phi_{n,m}(z,w)=[\pi_{n,m}(z,w),\Phi_{n,m-1}(z,w)^T]^T$. Hence from (3.16) $\mathcal{K}_{n,m}=0$. □

This leads to the following alternative proof of the two-variable Fejér–Riesz theorem in [8].

THEOREM 7.3. [illegible] $f(z,w)=\sum_{k=-n}^{n}\sum_{l=-m}^{m} f_{kl}z^kw^l$ [illegible] $|z|=|w|=1$ [illegible]

$$p(z,w)=\sum_{k=0}^{n}\sum_{l=0}^{m} p_{kl}z^kw^l,$$

[illegible] $p(z,w)\neq 0$ [illegible] $|z|,|w|\le 1$ [illegible] $f(z,w)=|p(z,w)|^2$ [illegible] $\mathcal{K}_{n,m}=0$

[illegible]. For $g\in C(\mathbb{T}^2)$ let $\mathcal{L}(g)=\frac{1}{(2\pi)^2}\int_{\mathbb{T}^2}\frac{g(\theta,\phi)}{|p(e^{i\theta},e^{i\phi})|^2}d\theta d\phi$. Then $\mathcal{L}$ is a positive definite linear functional on $\mathbb{T}^2$. The necessary part of the above result now follows from Theorem 7.2. The sufficiency also follows from the above theorem and the maximal entropy condition [1]. □

An alternative approach for finding a factorization as above may be done using the notion of intersecting zeros (see [9]). Also, the question of factorizing a nonnegative trigonometric polynomial as a modulus square of an outer polynomial was addressed in [6], allowing for generalizations in the operator valued case. When such a factorization of the desired degree does not exist, one can approximate the trigonometric polynomial with one that does have the desired factorization. This question was pursued in [12].

The vanishing of $\mathcal{K}_{n,m}$ has the following geometric interpretation.

LEMMA 7.4. [illegible] $\mathcal{L}$ [illegible] $\prod^{n,m}$ [illegible] $\mathcal{K}_{n,m}=0$ [illegible] $\Phi_{n,m-1}$ [illegible] (3.3)

$$\langle\Phi_{n,m-1}, z^iw^m\rangle=0,\quad 0\le i\le n-1. \tag{7.3}$$

[illegible]. The definition of $\Phi_{n,m-1}$ shows that it is already orthogonal to $z^iw^j$, $0\le i<n,\ 0\le j\le m-1$. The remaining orthogonality conditions show that $\Phi_{n,m-1}$ is orthogonal to all of the monomials in $\tilde{\Phi}_{n-1,m}$. Thus the sufficiency part of the theorem follows from (3.22). To see the necessary part note that from the definition of $\Phi_{n,m-1}$

$$\mathcal{K}_{n,m}=\left\langle\Phi_{n,m-1},\tilde{\Phi}^{n-1}_{n-1,m}\begin{bmatrix}z^{n-1}\\ \vdots\\ 1\end{bmatrix}w^m\right\rangle, \tag{7.4}$$

with $\tilde{\Phi}^{n-1}_{n-1,m}$ an invertible matrix. Thus (7.3) follows. □

Unfortunately at this point we are unable to see what the condition $\mathcal{K}_{n,m}=0$ implies for $u_{i,j},\ |i|\le n,\ |j|\le m$, except for $u_{n,-m}=0$, which follows from (5.4). We can, however, get a partial characterization for when a positive measure on the bicircle can be written as the reciprocal of the magnitude square of a stable polynomial. We begin with the following auxiliary result.

LEMMA 7.5. [illegible] $\hat{E}_{i,j}=0$ [illegible] $K^1_{i,j}$ [illegible] $u_{i,j}=0$ [illegible] $\hat{E}_{i,j}$, $\mathcal{K}_{i,j}$ [illegible] $\mathcal{K}_{i-1,j}$ [illegible] $\hat{E}_{i,j-1}$ [illegible] $\mathcal{K}_{i,j}$, $K_{i-1,j}$ $\hat{E}_{i,j-1}$ [illegible] $u_{i,j}$ [illegible] $\hat{E}_{i,j}=0$ [illegible] $\mathcal{K}^1_{i,j}=[0,\mathcal{K}^1_{i-1,j}]$ [illegible] $\hat{\hat{E}}_{i,j}=0$ [illegible] $K^1_{i,j}$ [illegible] $\hat{\hat{E}}_{i,j}$, $\mathcal{K}_{i,j}$ [illegible] $\mathcal{K}_{i,j-1}$ [illegible] $\hat{\hat{E}}_{i,j-1}$ [illegible] $\mathcal{K}_{i,j}$, $K_{i,j-1}$ $\hat{E}_{i-1,j}$ [illegible] $u_{i,j}$ [illegible] $\hat{\hat{E}}_{i,j}=0$ [illegible] $\mathcal{K}^1_{i,j}=[0,(\mathcal{K}^1_{i,j-1})^T]^T$

[illegible] If $\hat{E}_{i,j}=0$, then (3.51) and Remark 3.9 show that the first column of $\mathcal{K}^1_{i,j}$ is zero. If $\mathcal{K}_{i-1,j}$ is equal to zero, then (3.54) shows that all of the entries of $I_{i-1,j}$ are zero except for a one in the first entry. Thus (3.50) and (3.51) imply that if $\hat{E}_{i,j}=0$, $\mathcal{K}_{i,j}=0$, and $\mathcal{K}_{i-1,j}=0$, then $\hat{E}_{i,j-1}\bar{\Gamma}^1_{i-1,j}=0$ and $\hat{E}_{i,j-1}\bar{\Gamma}_{i-1,j}=0$. Following the argument in the construction of $\hat{E}_{n,m}$ we see that $\hat{E}_{i,j-1}=0$. The above hypothesis on $\mathcal{K}_{i-1,j}$ shows that $\tilde{\Gamma}_{i-1,j}=U_{i-1}$; thus (5.6) and the fact that the first column of $\mathcal{K}^1_{i,j}$ is zero gives $\mathcal{K}^1_{i,j}=[0,\mathcal{K}^1_{i-1,j}]$. The converse statement follows from (5.8). The remaining statements follow in an analogous fashion using Proposition 3.8. ☐

LEMMA 7.6. [illegible] $\mu$ [illegible] $\mu$ [illegible]

$$d\mu(\theta,\phi)=\frac{1}{|p_{n,m}|^2}d\theta d\phi,$$

[illegible] $p_{n,m}$ [illegible] $(n,m)$ [illegible] $\overleftarrow{p}_{n,m}(z,w)$ [illegible]

$$\mathcal{K}_{i,j}=0,\ \hat{E}_{i+1,j}=0,\ \text{[illegible]}\ \hat{\hat{E}}_{n,j+1}=0,\ i\ge n, j\ge m.$$

[illegible] Suppose that $d\mu=\frac{1}{|p_{n,m}(z,w)|^2}d\theta d\phi$, with $\overleftarrow{p}_{n,m}$ stable; then the sequence

$$\{\psi_{i,j}(z,w)\}\quad \psi_{i,j}(z,w)=z^{i-n}w^{j-m}p_{n,m}(z,w),\ i\ge n,\ j\ge m,$$

is a set of polynomials with degrees $(i,j)$, respectively, such that $\overleftarrow{\psi}_{i,j}=\overleftarrow{p}_{n,m}$ are stable and have the spectral matching property. Thus Theorem 7.2 implies that $\mathcal{K}_{i,j}=0$ for $i\ge n,\ j\ge m$. Since $\overleftarrow{\psi}_{i+1,j}=\overleftarrow{\psi}_{i,j}$, $i\ge n,\ j\ge m$, we see from (2.1.5) in [8] that $\overleftarrow{\Phi}_{i+1,j}=\overleftarrow{\Phi}_{i,j}$ for $i\ge n,\ j\ge m$. This implies that $A_{i+1,j}=I_{j+1}$ so that (3.14) shows that $\hat{E}_{i+1,j}=0$, $i\ge n, j\ge m$. Since $\overleftarrow{\tilde{\phi}}{}^i_{i,j+1}=\overleftarrow{\tilde{\phi}}{}^i_{i,j}$, $i\ge n,\ j\ge m$, the preceding argument shows that $\tilde{E}_{i,j+1}=0$, $i\ge n, j\ge m$. This proves the necessary part.

To prove sufficiency note that if $\mathcal{K}_{i,j}=0, i\ge n,\ j\ge m$, there exist polynomials $\psi_{i,j}$ of degree $(i,j)$ where $\overleftarrow{\psi}_{i,j}$ is a stable polynomial which has the spectral matching property. In order to show that $\overleftarrow{\psi}_{i,j}=\overleftarrow{\psi}_{n,m}$ we note that since $\hat{E}_{i+1,j}=0$ (3.14) implies that $\Phi_{i+1,j}=\Phi_{i,j}$, $i\ge n,\ j\ge m$. Furthermore $\hat{\hat{E}}_{n,j+1}=0$, $j\ge m$, implies that $\tilde{\Phi}_{n,j+1}=\tilde{\Phi}_{n,m}$. Since $\psi_{i,j}=\phi^j_{i,j}=\tilde{\phi}^i_{i,j}$ for $i\ge n,\ j\ge m$, the result follows. ☐

The conditions on $\mathcal{K}_{i,j}$, $E_{i,j}$, and $\tilde{E}_{n,j}$ given in Lemma 7.6 are not optimal since they are redundant. Some of this redundancy is removed in the next theorem.

THEOREM 7.7. [illegible] $\mu$ [illegible] $\mu$ [illegible] $d\mu=\frac{d\theta d\phi}{|p_{n,m}|^2}$ [illegible] $p_{n,m}$ [illegible] $(n,m)$ [illegible] $\overleftarrow{p}_{n,m}$ [illegible]

(a) $\mathcal{K}_{n,j} = 0$ $\tilde{\hat{\hat{E}}}_{n-1,j+1} = 0$ [illegible] $u_{n,j+1} = 0$ $j \geq m$.
(b) $\mathcal{K}_{i,m} = 0$ $\hat{E}_{i,m-1} = 0$ [illegible] $u_{i,m} = 0$ $i > n$.
(c) $u_{|i|,j} = 0, i > n,\ j > m$

[illegible] 7.8. Equation (5.4) and Lemma 7.5 show that $u_{-n,j}$, $u_{-i,m}$, $u_{n-1,j+1}$, and $u_{i,m-1}$ are also equal to zero for $j \geq m, i > n$.

[illegible] If $\mu$ has the form indicated in the hypotheses, then Lemma 7.6 says that $\mathcal{K}_{i,j} = 0,\ i \geq n,\ j \geq m$, which coupled with (5.4) implies that $u_{-i,j} = 0$, $i \geq n,\ j \geq m$; the remaining conditions on the coefficients follow from Lemma 7.5. If the coefficients obey (a)–(c), then Lemma 7.5 shows that $\hat{E}_{i+1,m}$ and $\hat{\hat{E}}_{n,j+1}$ are equal to zero for $i \geq n$ and $j \geq m$. Since $\hat{E}_{n,m+1} = 0$, $\hat{\hat{E}}_{n+1,m} = 0$, and by hypothesis $u_{-n-1,m+1} = 0$, (5.5) shows that $\mathcal{K}_{n+1,m+1} = 0$. With this Lemma 7.5 shows that $\hat{E}_{n+1,m+1} = 0$ and $\hat{\hat{E}}_{n+1,m+1} = 0$. The result now follows by induction. ☐

It is possible to modify slightly the hypotheses of Theorem 7.7 to obtain a statement just on the coefficients in the recurrence formulas.

THEOREM 7.9. [illegible] $u_{i,j}$ [illegible] (6.1) [illegible] $0 \leq i \leq n$, $|j| \leq m$ [illegible] 7.7 [illegible] $f \in C(\mathbb{T}^2)$

$$\mathcal{L}(f) = \left(\frac{1}{2\pi}\right)^2 \int_{\mathbb{T}^2} f(\theta, \phi) d\mu(\theta, \phi),$$

[illegible] $\mu$ [illegible] $d\mu = \frac{d\theta d\phi}{|p_{n,m}|^2}$ [illegible] $p_{n,m}$ [illegible] $(n, m)$ [illegible] $\overleftarrow{p}_{n,m}$ [illegible]

[illegible] From Theorem 6.1 there exists a positive definite linear functional on $\prod^{n,m}$ with the above parameters, and from Theorem 7.2 the functional has the representation

$$\mathcal{L}(e^{-ik\theta}e^{-il\phi}) = \left(\frac{1}{2\pi}\right)^2 \int_{\mathbb{T}^2} \frac{e^{-ik\theta}e^{-il\phi}}{|p_{n,m}(e^{i\theta}e^{i\phi})|^2} d\theta d\phi, \quad |k| \leq n,\ |l| \leq m,$$

with $p_{n,m}$ a polynomial of degree $(n, m)$ with $\overleftarrow{p}_{n,m}$ stable. The result now follows from Theorem 7.7. ☐

**8. Examples.** We now give some examples that illustrate various aspects of the results presented earlier. We begin with the case $n = 1, m = 1$, with $u_{0,0} = 1$, $\mathcal{K}_{1,1} = u_{-1,1} = 0$, and $\mathcal{K}^1_{1,1} = \bar{u}_{1,1}$. From Theorem 6.1 we see that we must choose $|u_{0,1}| < 1$ and $|u_{1,0}| < 1$. Since $\mathcal{K}_{1,1} = 0$ the only remaining condition for $\mathcal{L}$ to be a positive linear functional on $\prod^{1,1}$ is for $e^{1\,T}_1 \tilde{H}^3_{1,1} e^1_1 < 1$. From (5.9) we see that $\Gamma^1_{1,1} U^T_1 = I_{1,0} \hat{\hat{E}}_{1,1} (I^1_{1,0})^T$. The construction of $I_{1,0}$, $I^1_{1,0}$, and (5.8) shows that $e^{1\,T}_1 \tilde{H}^3_{1,1} e^1_1 < 1$ is given by

$$a|u_{1,1}|^2 + b(\bar{u}_{1,1}\bar{u}_{0,1}u_{1,0} + u_{1,1}u_{0,1}\bar{u}_{1,0}) + c < 1,$$

with $a = \frac{1-|u_{0,1}u_{1,0}|^2}{1-|u_{1,0}|^2}$, $b = \frac{\sqrt{1-|u_{0,1}|^2}}{\sqrt{1-|u_{1,0}|^2}}$, and $c = |u_{0,1}|^2$. This simplifies to

$$|\hat{u}_{1,1}| < 1,$$

where

$$\hat{u}_{1,1} = \frac{(1 - |u_{0,1}u_{1,0}|^2)u_{1,1}}{\sqrt{1 - |u_{0,1}|^2}\sqrt{1 - |u_{1,0}|^2}} + u_{0,1}\bar{u}_{1,0}.$$

Thus from Theorems 6.1 and 7.2 we see that with $u_{0,0} = 1$

$$\mathcal{L}(e^{-ik\theta}e^{-ij\phi}) = \left(\frac{1}{2\pi}\right)^2 \int_{\mathbb{T}^2} \frac{e^{-ik\theta}e^{-ij\phi}}{|\phi_{1,1}(e^{i\theta}, e^{i\phi})|^2} d\theta d\phi, \ |k| \le 1, \ |j| \le 1,$$

where $\phi_{1,1}$ constructed using (3.16) and the top row of (3.17) is a polynomial of degree (1,1) with $\overleftarrow{\phi}_{1,1}$ stable if and only if $|u_{0,1}| < 1$, $|u_{1,0}| < 1$, $u_{-1,1} = 0$, and $|\hat{u}_{1,1}| < 1$. Furthermore if we set $u_{j,0}$, $u_{0,j}$, $u_{i,j}$ equal to zero for $i > 1, |j| > 1$, then Theorem 7.9 shows that the above representation for $\mathcal{L}$ extends to all continuous functions on $\mathbb{T}^2$.

We can also use the previous results to investigate contractive Toeplitz matrices. In this case we find

$$C_{1,1} = \begin{bmatrix} I & C_{-1} \\ C_1 & I \end{bmatrix}, \tag{8.1}$$

where $C_{-1} = C_1^\dagger$ is a $2 \times 2$ Toeplitz matrix. In this case $u_{0,0} = 1$ and $u_{0,1} = 0$ so that $\tilde{E}_{0,1} = 0$, $\tilde{A}_{0,1} = 1$. Since $\mathcal{K}_{1,1} = u_{-1,1}$, we find $\Gamma_{1,1} = [0, \sqrt{1 - |u_{-1,1}|^2}]$. This plus the computation of $\tilde{\Gamma}^1_{1,0}$ described in the construction of $\mathcal{L}$ yields

$$\begin{aligned} I &= (e_1^1)^T \tilde{H}^3_{1,1} (\tilde{H}^3_{1,1})^\dagger (e_1^1) \\ &= (1 + d)|u_{1,1}|^2 + d(u_{1,1}u_{-1,1} + \bar{u}_{1,1}\bar{u}_{-1,1}) + d|u_{-1,1}|^2 < 1, \end{aligned} \tag{8.2}$$

where

$$d = \frac{|u_{1,0}|^2}{(1 - |u_{1,0}|^2)(1 - |u_{-1,1}|^2)}.$$

By completing the square this can be simplified to

$$|\hat{u}_{1,1}| < 1,$$

where

$$\hat{u}_{1,1} = (1 + d)\sqrt{1 - |u_{1,0}|^2} u_{1,1} + d_1 \bar{u}_{-1,1},$$

and

$$d_1 = \frac{|u_{1,0}|^2}{(1 - |u_{-1,1}|^2)\sqrt{1 - |u_{1,0}|^2}},$$

which puts constraints on $u_{-1,1}$. Thus we find that the conditions for $\mathcal{L}$ to be a positive linear functional and hence $C_1$ to be a contractive Toeplitz matrix are $|u_{1,0}| < 1$, $|u_{-1,1}| < 1$, and $|\hat{u}_{1,1}| < 1$. These constraints may not be strong enough to allow $\mathcal{L}$ to be extended. To see this suppose $n = 1, m = 2$, $u_{0,2} = 0$, and $u_{1,0} = 0$. It is not difficult to see then that $\hat{E}_{1,1} = \operatorname{diag}(\bar{u}_{1,1}, u_{-1,1})$. With $u_{1,0} = 0$ the constraint on $\hat{u}_{1,1}$ above reduces to $|u_{1,1}| < 1$. However,

$$K_{1,2} = \begin{pmatrix} \frac{u_{-1,1}}{(1-|u_{1,1}|^2)^{1/2}} \\ \frac{u_{-1,2}}{(1-|u_{-1,1}|^2)^{1/2}} \end{pmatrix},$$

so we see that in order for $K_{1,2}$ to be a contraction $\frac{|u_{-1,1}|}{\sqrt{1-|u_{1,1}|^2}} < 1$, which may not be satisfied.

## REFERENCES

[1] M. Bakonyi and G. Naevdal, *On the matrix completion method for multidimensional moment problems*, Acta Sci. Math. (Szeged), 64 (1998), pp. 547–558.

[2] A. M. Delgado, J. S. Geronimo, P. Iliev, and F. Marcellán, *Two variable orthogonal polynomials and structured matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 118–147.

[3] Ph. Delsarte, Y. V. Genin, and Y. G. Kamp, *Orthogonal polynomial matrices on the unit circle*, IEEE Trans. Circuits Syst. I Regul. Pap., 25 (1978), pp. 149–160.

[4] Ph. Delsarte, Y. V. Genin, and Y. G. Kamp, *Planar least squares inverse polynomials.* I. *Algebraic properties*, IEEE Trans. Circuits Syst. I Regul. Pap., 26 (1979), pp. 59–66.

[5] M. A. Dritschel, *On factorization of trigonometric polynomials*, Integral Equations Operator Theory, 49 (2004), pp. 11–42.

[6] M. A. Dritschel and H. J. Woerdeman, *Outer factorizations in one and several variables*, Trans. Amer. Math. Soc., 357 (2005), pp. 4661–4679.

[7] Y. V. Genin and Y. G. Kamp, *Two-dimensional stability and orthogonal polynomials on the hypercircle*, Proc. IEEE, 65 (1977), pp. 873–881.

[8] J. S. Geronimo and H. J. Woerdeman, *Positive extensions, Fejér-Riesz factorization and autoregressive filters in two variables*, Ann. of Math., 160, (2004), pp. 839–906.

[9] J. S. Geronimo and H. J. Woerdeman, *Two-variable Polynomials: Intersecting zeros and stability*, IEEE Trans. Circuits Syst. I Regul. Pap., 53, (2005), pp. 1130-1139.

[10] J. S. Geronimo and M. J. Lai, *Factorization of multivariate Laurent polynomials*, J. Approx. Theory, 139 (2006), pp. 327–345.

[11] I. Gohberg and G. Heinig, *Inversion of finite Toeplitz matrices consisting of elements of a noncommutative algebra*, Rev. Roumaine Math. Pures Appl., 19 (1974), pp. 623–663.

[12] Y. Hachez and H. J. Woerdeman, *Approximating sums of squares with a single square*, Linear Algebra Appl., 399 (2005), pp. 87–201.

[13] H. Helson, *Lectures on Invariant Subspaces*, Academic, New York, 1964.

[14] D. Jackson, *Formal properties of orthogonal polynomials in two variables*, Duke Math. J., 2 (1936), pp. 423–434.

[15] T. Kailath, A. Vieira, and M. Morf, *Inverses of Toeplitz operators, innovations and orthogonal polynomials*, SIAM Rev., 20 (1978), pp. 106–119.

[16] J. W. Mclean and H. J. Woerdeman, *Spectral factorization and sums of squares representations via semidefinite programming* SIAM J. Matrix Anal. Appl., 23 (2001), pp. 646–655.

[17] L. Rodman, *Orthogonal matrix polynomials*, Orthogonal Polynomials, NATO Sci. Ser. C Math. Phys. Sci. 294, Paul Nevai, ed., Kluwer Academic, Dordrecht, 1990, pp. 345–362,

[18] W. Rudin, *Fourier analysis on groups.* in Interscience Tracts in Pure and Applied Math. 12, L. Bers, Interscience, NY, 1962.

[19] B. Simon, *Orthogonal polynomials on the unit circle. Part* 1. *Classical theory*, in Amer. Math. Soc. Colloq. Publ. 54, Part 1. American Mathematical Society, Providence, RI, 2005.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 826–837

# PARALLEL BIDIAGONALIZATION OF A DENSE MATRIX*

CARLOS CAMPOS[†], DAVID GUERRERO[‡], VICENTE HERNÁNDEZ[‡], AND RUI RALHA[§]

**Abstract.** A new stable method for the reduction of rectangular dense matrices to bidiagonal form has been proposed recently. This is a one-sided method since it can be entirely expressed in terms of operations with (full) columns of the matrix under transformation. The algorithm is well suited to parallel computing and, in order to make it even more attractive for distributed memory systems, we introduce a modification which halves the number of communication instances. In this paper we present such a modification. A block organization of the algorithm to use level 3 BLAS routines seems difficult and, at least for the moment, it relies upon level 2 BLAS routines. Nevertheless, we found that our sequential code is competitive with the LAPACK DGEBRD routine. We also compare the time taken by our parallel codes and the ScaLAPACK PDGEBRD routine. We investigated the best data distribution schemes for the different codes and we can state that our parallel codes are also competitive with the ScaLAPACK routine.

**Key words.** bidiagonal reduction, parallel algorithms

**AMS subject classifications.** 15A18, 65F30, 68W10

**DOI.** 10.1137/05062809X

## 1. Introduction.

The problem of computing the singular value decomposition (SVD) of a matrix is one of the most important operations in numerical linear algebra and is employed in a variety of applications. The SVD is defined as follows.

For any rectangular matrix $A \in \mathbb{R}^{m\times n}$ (we will assume that $m \geq n$), there exist two orthogonal matrices $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ and a matrix $\Sigma = \left[\begin{smallmatrix}\Sigma_A\\0\end{smallmatrix}\right] \in \mathbb{R}^{m\times n}$, where $\Sigma_A = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ is a diagonal matrix, such that $A = U\Sigma V^t$. The values $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ are called the singular values of $A$.

To compute the SVD of a dense matrix, an important class of methods starts with [illegible] [11], [12], [14], [15], [19], [20], [21], reducing $A$ to upper bidiagonal form $B \in \mathbb{R}^{n\times n}$, from which the singular values are computed in an iterative manner [2], [16], [18], [22].

The Householder bidiagonalization computes $U_B \in \mathbb{R}^{m\times m}$ and $V_B \in \mathbb{R}^{n\times n}$, as products of Householder matrices, such that

$$A = U_B \begin{bmatrix} B \\ 0 \end{bmatrix} V_B^t, \qquad \text{where} \qquad B = \begin{bmatrix} \alpha_1 & \beta_2 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \beta_n \\ & & & \alpha_n \end{bmatrix}. \tag{1}$$

The classical method for producing this decomposition is a two-sided algorithm which employs both premultiplication and postmultiplication by Householder matri-

---

*Received by the editors March 30, 2005; accepted for publication (in revised form) by J. L. Barlow February 1, 2007; published electronically July 25, 2007.
http://www.siam.org/journals/simax/29-3/62809.html

[†] Departamento de Matemática, Escola Superior de Tecnologia e Gestão (Leiria), Campus 2, Morro do Lena, Alto do Vieiro, 2401-951 Leiria, Portugal (ccampos@estg.ipleiria.pt).

[‡] Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Camino de Vera s/n, E-46022 Valencia, Spain (dguerrer@dsic.upv.es, vhernand@dsic.upv.es).

[§] Departamento de Matemática, Universidade do Minho, 4710-057 Braga, Portugal (r_ralha@math.uminho.pt). The research of this author was supported by the Portuguese Foundation for Science and Technology through the research program POCI 2010.

ces. In order to establish an algorithm which is better suited to parallel implementation than the standard bidiagonalization method, Ralha [25], [26], and Ralha and Mackiewicz [27] proposed a new technique that uses only multiplication on the right side of $A$ by Householder matrices. Inspired by Ralha's method, Barlow, Bosner, and Drmač [4] proposed a new stable method for the reduction of rectangular dense matrices to the bidiagonal form.

In this paper we present both methods and propose a modification to the Barlow method which halves the number of communication instances in the parallel implementation, making the algorithm even more attractive for distributed memory systems.

This paper is organized as follows. In section 2 we describe the new bidiagonalization methods. Section 3 deals with the sequential and parallel implementations, using LAPACK [1] and ScaLAPACK [8] routines. In section 4 we analyze the experimental results of our numerical tests. Section 5 summarizes our conclusions and future work.

## 2. New bidiagonalization methods.

### 2.1. Ralha bidiagonal reduction.

Given a rectangular dense matrix $A \in \mathbb{R}^{m\times n}$, the bidiagonalization method proposed by Ralha is comprised of two stages. The first stage consists of a sequence of $n-2$ Householder transformations

$$A_r = A_{r-1} \cdot \mathrm{diag}\,(I_r, H_r) \qquad (r = 1, \ldots, n-2), \tag{2}$$

where $I_r$ is the identity matrix of order $r$, $A_0 = A$, and the columns $a_i$ and $a_j$ of the final matrix $A_{n-2}$ satisfy

$$a_i^t a_j = 0 \qquad \text{for} \qquad |i-j| > 1. \tag{3}$$

This can be understood as an implicit reduction of the symmetric semidefinite positive matrix $A^tA$ to tridiagonal form. In the $r$th step, the construction of the Householder vector $v_r$ in

$$H_r = I_{n-r} - \frac{2}{v_r^t v_r} v_r v_r^t \tag{4}$$

requires the computation of $n-r$ dot products involving the appropriate columns of $A_{r-1}$.

Having produced $A_{n-2}$, the second stage is a variant of the Gram–Schmidt orthogonalization method that produces the factorization $A_{n-2} = QB$, where $B$ is the required upper bidiagonal matrix.

Representing by $a_i$ and $q_i$ the columns of $A_{n-2}$ and $Q$, respectively, we have

$$\begin{bmatrix} a_1 \cdots & a_i \cdots & a_n \end{bmatrix} = \begin{bmatrix} q_1 \cdots & q_i \cdots & q_n \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ & \alpha_2 & \ddots & & & \\ & & \ddots & \beta_i & & \\ & & & \alpha_i & \ddots & \\ & & & & \ddots & \beta_n \\ & & & & & \alpha_n \end{bmatrix} \tag{5}$$

with

$$q_1 = \frac{a_1}{\alpha_1}, \qquad q_i = \frac{a_i - \beta_i q_{i-1}}{\alpha_i} \quad (i = 2, \ldots, n)\,. \tag{6}$$

Each $\beta_i$ is chosen to make $q_i$ orthogonal to $q_{i-1}$, and each $\alpha_i$ is such that $\|q_i\|_2 = 1$; with these conditions, we get from (6)

$$\begin{aligned} \alpha_1 &= \|a_1\|_2\,, \\ \beta_i &= a_i^t q_{i-1} \quad (i = 2, \ldots, n)\,, \\ \alpha_i &= \|a_i - \beta_i q_{i-1}\|_2 \quad (i = 2, \ldots, n)\,. \end{aligned} \tag{7}$$

The first stage of this method is perfectly stable in the sense that the computed $\widetilde{A}_{n-2}$ satisfies

$$\widetilde{A}_{n-2} = (A + E)P, \tag{8}$$

where $P$ is exactly orthogonal and

$$\|E\|_2 \leq g(m, n)\varepsilon_M \|A\|_2 \tag{9}$$

for some modestly growing function $g(m, n)$ and machine epsilon $\varepsilon_M$ [24, pp. 94–96]. Furthermore, if $A = DX$, where $D$ is diagonal and $cond(X) \ll cond(A)$, then these one-sided orthogonal transformations preserve the small singular values better than two-sided transformations [10].

It may happen that some nonadjacent columns of $\widetilde{A}_{n-2}$ are not orthogonal to working precision[1] and, even when all those columns are numerically orthogonal, the process of producing a bidiagonal $B$ from $\widetilde{A}_{n-2}$ may bring trouble. To give an insight into the problem, consider the following triangular matrix:

$$R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 10^{-9} & 1 & 10^{-9} & 10^{-7} \\ 0 & 0 & 10^{-3} & 10^{-6} & -10^{-4} \\ 0 & 0 & 0 & 1 & 10^{-11} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We have

$$R^t R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 + 10^{-18} & 10^{-9} & 10^{-18} & 10^{-16} \\ 0 & 10^{-9} & 1 + 10^{-6} & 2 \times 10^{-9} & 0 \\ 0 & 10^{-18} & 2 \times 10^{-9} & 1 + O(10^{-12}) & O(-10^{-10}) \\ 0 & 10^{-16} & 0 & O(-10^{-10}) & 1 + O(10^{-8}) \end{bmatrix},$$

which differs from a tridiagonal matrix by quantities not larger than $10^{-16}$. If $A = QR$, with $Q$ orthogonal, then $A^t A = R^t R$ and the nonadjacent columns of $A$ are orthogonal to working precision; however, from the uniqueness of the QR decomposition, it follows that there is no bidiagonal $B$ satisfying $A = QB$; in other words, small perturbations outside the tridiagonal band of $A^t A$ cause much larger (as large as $O(10^{-4})$ in this case) perturbations in $B$; i.e., the problem is ill-conditioned.

---

[1] See [26] for an example: the Lauchli matrix $L(n, \mu)$ with $n = 7$ and $\mu = \varepsilon_M$.

**2.2. Barlow bidiagonal reduction.** Barlow, Bosner, and Drmač [4] recently proposed a stable algorithm which consists, essentially, in interleaving the two stages of Ralha's method. More precisely,

1. the vector $q_r$ is computed immediately after applying the Householder transformation $H_r$; and
2. the Householder vector used in the transformation $H_{r+1}$ is computed from $x_r = A\left(:, r+1:n\right)^t q_r$.

Those authors also noted that the computation of the dot product $\beta_r = q_{r-1}^t A\left(:, r\right)$ may be avoided (this is also the case with Ralha's algorithm); in [4], an extensive error analysis of the proposed algorithm is carried out which shows that the method is always able to compute an upper bidiagonal matrix $B$ in a backward stable manner. That is, we have, for each $k = 1, \ldots, n$,

$$|\sigma_k(B) - \sigma_k(A)| \leq f(m,n)\varepsilon_M \left\|A\right\|_2 + O\left(\varepsilon_M^2\right) \tag{10}$$

for some modestly growing function $f(m,n)$. As was already the case with Ralha's bidiagonalization, there are ill-conditioned matrices where this algorithm obtains small relative errors for all the singular values [26], [10]. Furthermore, the orthogonality of the columns of $U$ is similar to that of the matrix $Q$ in the QR factorization by the modified Gram–Schmidt method [6], [7]; that is, $U$ may be far from orthogonal (see [4, Example 3.1]). Nevertheless, the leading left singular vectors of $A$ can be recovered with good orthogonality (see [4, Corollary 3.20]).

The algorithm may be stated simply as follows (for a more complete statement see [4]).

Algorithm 1 (Barlow bidiagonal reduction).

```
[illegible] $r = 1 : n-2$
    $\alpha_r = \left\|A\left(:, r\right)\right\|_2$
    $q_r = \frac{A(:,r)}{\alpha_r}$
    $x_r = A\left(:, r+1:n\right)^t q_r$
    [illegible] $H_r$ [illegible] $H_r^t x_r = \beta_{r+1} e_1$
    $A\left(:, r+1:n\right) = A\left(:, r+1:n\right) H_r$
    $A\left(:, r+1\right) = A\left(:, r+1\right) - \beta_{r+1} q_r$

[illegible]
$\alpha_{n-1} = \left\|A\left(:, n-1\right)\right\|_2$
$q_{n-1} = \frac{A(:,n-1)}{\alpha_{n-1}}$

$\beta_n = q_{n-1}^t A\left(:, n\right)$
$A\left(:, n\right) = A\left(:, n\right) - \beta_n q_{n-1}$
$\alpha_n = \left\|A\left(:, n\right)\right\|_2$
$q_n = \frac{A(:,n)}{\alpha_n}$
```

**2.3. Modified Barlow bidiagonal reduction.** The advantages of one-sided transformations for parallel bidiagonalization on a multiprocessor system with distributed memory were first discussed in [25]. The best data decomposition consists in assigning to each one of $p$ processors a number of rows of the matrix under transformation, that is, a segment of length $m/p$ of each column (we are assuming, for simplicity, that $p$ is a divisor of $m$; if this is not the case, then each processor should get either $floor(m/p)$ or $floor(m/p) + 1$ rows). According to the ideas proposed in [25], in the $r$th step, each processor gets a copy of the entire vector $x_r$ and computes its own copy of the corresponding Householder vector. So, there is some redundancy in the computation, but its negative effect in the overall efficiency is not dramatic when $m/p$ is large (see [25]). The reward for this approach is two-fold: the load balancing is optimal and interprocessor communication is required only for the $n-r+1$ dot products involved in the computation of the norm $\alpha_r$ and the vector $x_r$.

Following this strategy, in the parallel implementation of Barlow's method, a communication event, involving all processors, is required to compute $\alpha_r$ alone. Then, a normalization is carried out to produce $q_r$, and this will be followed by a second communication event involving the processors in the computation of the $n-r$ dot products $x_r = A\left(:, r+1:n\right)^t q_r$.

These two communication events may be reduced to only one if we postpone the normalization of the $r$th column of $A_r$; that is, the processors cooperate in the global task of computing the $n-r+1$ dot products $x_r = A\left(:, r:n\right)^t A\left(:, r\right)$ and, from these, each processor will compute locally $\alpha_r = \sqrt{x_r\left(1\right)}$, where $x_r\left(1\right) = A\left(:, r\right)^t A\left(:, r\right)$, and the local segment of $q_r = A\left(:, r\right)/\alpha_r$. Observe that $x_r(2:n-r+1)$ differs from the vector $x_r$ computed in Barlow's method by a factor equal to $\alpha_r$, but there is no need to perform a scaling of $x_r(2:n-r+1)$ since the resulting Householder reflector in (4) is invariant under such a scaling. However, in the computation of the off-diagonal element we must take into account that the relation

$$\beta_{r+1} = \frac{\|x_r(2:n-r+1)\|_2}{\alpha_r} \tag{11}$$

holds in each step.

An essential ingredient in the proof presented in [4] for the error bound given in (10) is the fact that, in each step, the computed $A_r$ is the exact product $(A_{r-1}+E_r)\cdot \operatorname{diag}\left(I_r, H_r\right)$, where $H_r$ is the exact Householder reflector corresponding to the vector $x_r$ of the inner products and $\|E_r\|_2 \leq \quad (\varepsilon_M)\,\|A_{r-1}\|_2$. This matrix $E_r$ encapsulates the errors in the approximation $\hat{v}_r$ computed for the exact vector $v_r$ and also the errors produced in the update $A_{r-1}\cdot \operatorname{diag}(I_r, \hat{H}_r)$, with $\hat{H}_r = I_{n-r} - \frac{2}{\hat{v}_r^t\hat{v}_r}\hat{v}_r\hat{v}_r^t$. In the modified method, a slightly different approximation $\tilde{v}_r$ will be produced (for a detailed error analysis in the computation of the Householder vector see [30, pp. 152–157]), but, similarly to $\hat{v}_r$, $\tilde{v}_r$ defines a Householder reflector, say, $\tilde{H}_r$, that is very close to the exact one; i.e., we have $\|\tilde{H}_r - H_r\|_2 = \quad (\varepsilon_M)$. We therefore claim that the error analysis given in [4] also applies to our modified method.

Our proposal does not change the arithmetic complexity of Barlow's method and does not reduce the volume of data to be transferred but halves the number of communication events, therefore reducing the overhead caused by the latency in the communications. The total cost of communication depends upon the parallel computation of the inner products only; in [25] it is shown that, on a simple chain of processors, this cost is approximately given by

$$p\frac{n^2}{2}\left(t_{flop} + 2t_{com}\right), \tag{12}$$

where $t_{flop}$ represents the time taken by one floating point operation and $t_{com}$ stands for the time required to pass a floating point number from one processor to another. The factor $p$ in (12) essentially reflects the diameter of the network and may be replaced by $\sqrt{p}$ in the case of a square grid.

The computation of $\alpha_{n-1}$ and $\beta_n$ may also be arranged in a way that saves one communication event in the parallel implementation. As in the previous steps, we may use communication to get $x_{n-1} = A\left(:, n-1:n\right)^t A\left(:, n-1\right)$ in each processor;

then, we have

$$\alpha_{n-1} = \sqrt{x_{n-1}(1)} \qquad \text{and} \qquad \beta_n = x_{n-1}(2)/\alpha_{n-1}. \tag{13}$$

In practical implementations of these algorithms, $q_r$ may overwrite $A(:,r)$ to reduce the volume of the storage required. In the next section we do so.

## 3. Sequential and parallel implementations.

In this section we describe the methodology used to develop our sequential and parallel implementations. The same methodology was applied to all implementations, but from now on we will refer only to the sequential and parallel implementations of the modified Barlow method.

In order to obtain high portability and efficiency, all our implementations use, as much as possible, LAPACK and ScaLAPACK routines. It must be stressed that our implementations rely on level 2 BLAS routines [17]. A block organization of the Barlow method has been under development by Bosner and Barlow (see [9]), who have reported significant reduction in the execution time of the sequential algorithm, depending upon the size of the matrices. However, those authors also found that for parallel processing the nonblocked algorithm is preferred due to large overheads in the block version.

From our sequential codes we obtained the corresponding parallel codes by translating the BLAS and the LAPACK routines into calls of the equivalent parallel routines of PBLAS [13] and ScaLAPACK. This translation process takes into account the data distribution and the corresponding rules to convert sequential LAPACK-based programs into parallel ScaLAPACK-based programs.

Our parallel implementation of the modified Barlow method, including the corresponding PBLAS and ScaLAPACK routines, is stated as follows.

Algorithm 2 (parallel implementation).

```
[illegible]                                                        [illegible]
[illegible] $x_r$                                                  [illegible]
[illegible] $r = 1 : n-2$
    $x_r = A(:,r:n)^t A(:,r)$                                      [illegible]
    $\alpha_r = \sqrt{x_r(1)}$
    $A(:,r) = \frac{A(:,r)}{\alpha_r}$                             [illegible]
    [illegible] $H_r$ [illegible] $H_r^t x_r(2:n-r+1) = \phi_r e_1$    [illegible]
    $A(:,r+1:n) = A(:,r+1:n) H_r$                                  [illegible]
    $\beta_{r+1} = \frac{\phi_r}{\alpha_r}$
    $A(:,r+1) = A(:,r+1) - \beta_{r+1} A(:,r)$                     [illegible]
[illegible]
$x_{n-1} = A(:,n-1:n)^t A(:,n-1)$                                  [illegible]
$\alpha_{n-1} = \sqrt{x_{n-1}(1)}$
$\beta_n = x_{n-1}(2)/\alpha_{n-1}$
$A(:,n-1) = \frac{A(:,n-1)}{\alpha_{n-1}}$                         [illegible]
$A(:,n) = A(:,n) - \beta_n A(:,n-1)$                               [illegible]
$\alpha_n = \|A(:,n)\|_2$                                          [illegible] 2
$A(:,n) = \frac{A(:,n)}{\alpha_n}$                                 [illegible]
```

With the ScaLAPACK data distribution, which follows a two-dimensional block cyclic scheme, we manage to assign $m/p$ rows (not contiguous) to each processor by reducing the grid to a single column of processors. We emphasize that, as a direct consequence of the use of the routines from ScaLAPACK, we have not fully implemented the parallel algorithm as presented in the previous section. In our implementation, the computation of the inner products is carried out with PxGEMV and, as follows from the array descriptor that we have used, the resulting vector $x_r$ is stored on a single processor (processor 0, say). As a consequence of this distribution, during the execution of PxLARFG, the computation of $\phi_r$ is carried out on processor 0 only and no communication is required. The application of the Householder reflectors (with PxLARF) requires communication to make the value $\phi_r$ available on each processor.

Finally, we note that, in applications where it is not necessary to produce a matrix $Q$ with normalized columns, we may change Algorithm 2 in a way that reduces the number of floating point divisions. This consists of removing the scaling operations $A(:,r) = A(:,r)/\alpha_r$ (PxSCAL) for $r = 1, \ldots, n$ and rewriting the PxAXPY operations as $A(:,r+1) = A(:,r+1) - \frac{\beta_{r+1}}{\alpha_r} A(:,r)$ for $r = 1, \ldots, n-1$, with a total savings of $mn - (n-1)$ divisions.

## 4. Experimental results.

**4.1. Introduction.** In this section we analyze the execution times of our implementations, obtained on a cluster with 20 biprocessor nodes where each node is a Pentium Xeon at 2GHz, 1GB of RAM, and Redhat Linux operating system. The nodes are connected through a SCI network, organized in a $4 \times 5$ 2D torus grid. Each node has been treated as a single processor machine and the biprocessor feature has not been exploited. Unfortunately, only 10 nodes of the cluster were available for our computational tests.

All experiments were performed using Fortran 90 and IEEE standard double precision floating point arithmetic [23]. As already said, we made use of LAPACK and ScaLAPACK routines in order to ensure a high level of portability and efficiency of our implementations. The communications in ScaLAPACK were carried out using Scali MPI [28], which is an optimized implementation of the standard MPI communication library [29] for SCI networks.

In all experiments, the execution times were measured in seconds, and the test matrices (rectangular matrices with sizes ranging from $10000 \times 1000$ to $10000 \times 4500$ and square matrices with sizes ranging from $1000 \times 1000$ to $4500 \times 4500$) were generated randomly.

The execution times that will be reported are strictly for the process of producing the bidiagonal; i.e., no accumulation of the orthogonal transformations was carried out.

**4.2. Sequential codes.** In Figure 1 we compare the execution times of the LAPACK routine DGEBRD and our sequential codes for the case of rectangular matrices. If $m$ is much larger than $n$, it is more efficient to carry out an initial QR decomposition [11]. In our tests we have not done this, mainly because PDGEBRD (from ScaLAPACK) also does not perform such a decomposition. As can be seen, the new bidiagonalization methods have similar execution times, and, in general, our sequential codes are competitive with DGEBRD.

The number of flops involved in the new methods is approximately equal to $3mn^2$ flops and the operation count for DGEBRD is $4mn^2 - 4/3n^3$; therefore, the new methods require fewer flops whenever $m > \frac{4}{3}n$ [4], [25]. For $m$ fixed, the new methods are less competitive as $n$ grows. For $m = 10000$ and $n = 4000$, DGEBRD uses about $\frac{4}{3}$ the number of flops required by the new method. However, looking at Figure 1, we see that the execution times are almost equal. This is because DGEBRD applies block updates of the form $A - UX^t - YV^t$ using two calls to the level 3 BLAS routine DGEMM; these calls account for about half the work [1] and make the code more

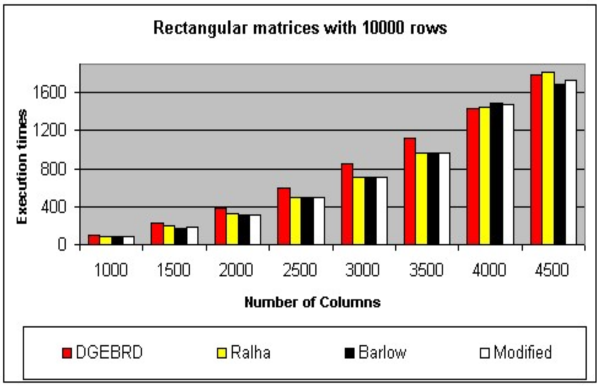


Fig. 1. *DGEBRD versus new methods (rectangular matrices).*

efficient. The new methods are based upon level 2 BLAS routines; i.e., the ratio of floating point operations to memory references is lower.

**4.3. Parallel codes.** In this section we compare, in terms of the execution times measured, the ScaLAPACK routine PDGEBRD and our parallel codes. We have observed that there is a nonnegligible influence of the sizes of the rectangular grid used for the configuration of the processors. Unlike the ScaLAPACK routine, our parallel implementations perform better on a grid with a single column. In the following comparisons we will always use the best execution time.

In Figure 2 we report the execution times of the Barlow and the modified Barlow parallel codes running on 2, 4, 6, 8, and 10 processors for rectangular matrices. For each $n$ (number of columns), there are five pairs of consecutive bars, one pair for each value of $p$ (number of processors). On each pair, the dark bar corresponds to the Barlow method, and the white bar corresponds to the modified method. The gain that can be observed for the parallel implementation of the modified method is not impressive. This is not surprising because our computational platform has efficient communication, as one can conclude from the high efficiency obtained for all the parallel algorithms. Since the modified method reduces the communication overheads, we expect the gain to be much more significant on a system where the cost of the communications is heavier (a loosely coupled network of personal computers, for example).

Figure 3 allows a comparison of the execution times of PDGEBRD and the modified method on 2, 4, 6, 8, and 10 processors. Again, for each $n$, there are five pairs of consecutive bars, dark and white, the dark bar corresponding to PDGEBRD, and the white bar to the modified method. As can be seen, on two processors our code is slower than PDGEBRD, but the situation is reversed as we increase the number of processors. At this point, in our experiments, we were very sorry to not have the opportunity to use many more processors since we do believe that the new method has better scalability than the ScaLAPACK routine. In section 5 we justify this conviction with some arguments.

In Figure 4, for each $n$, we give the efficiency obtained for PDGEBRD (dark) and for the parallel code of the modified Barlow algorithm (white), running on 2, 4, 6, 8,

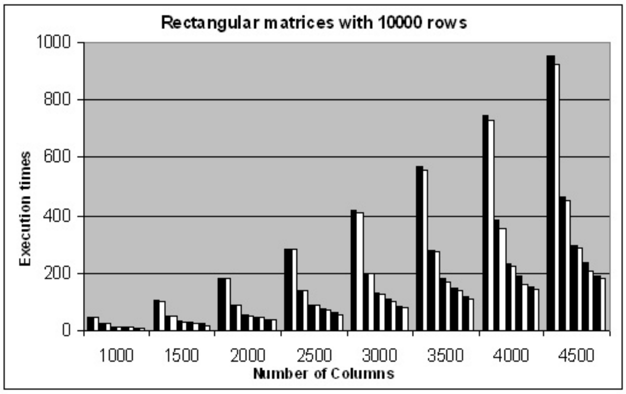


FIG. 2. *Barlow's versus modified (rectangular matrices).*

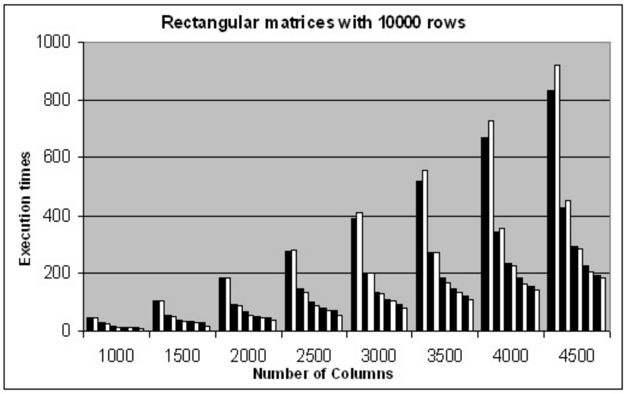


FIG. 3. *PDGEBRD versus modified (rectangular matrices).*

and 10 processors. The efficiency is computed according to the usual formula:

$$\text{Efficiency} = \frac{\text{Execution time on a single processor}}{p \times (\text{Execution time on } p \text{ processors})}. \tag{14}$$

Finally, to illustrate the influence of the processors grid, we ran PDGEBRD on a linear array of processors. This causes a significant degradation of the efficiency of PDGEBRD (compare the dark bars for efficiency in Figures 4 and 5), and, in this case, the new code is clearly more efficient.

**5. Conclusions and future work.** Inspired by an algorithm proposed by one of us, Barlow, Bosner, and Drmač recently presented a backward stable method for the reduction of a matrix to bidiagonal form.

We have presented parallel implementations of the method as proposed by those authors and also of a modified method that halves the number of communication events.

The advantages of one-sided transformations over two-sided methods for parallel bidiagonalization have been explained in detail in [25]. The parallel code for the one-sided algorithm is essentially the sequential code with a procedure to compute the dot

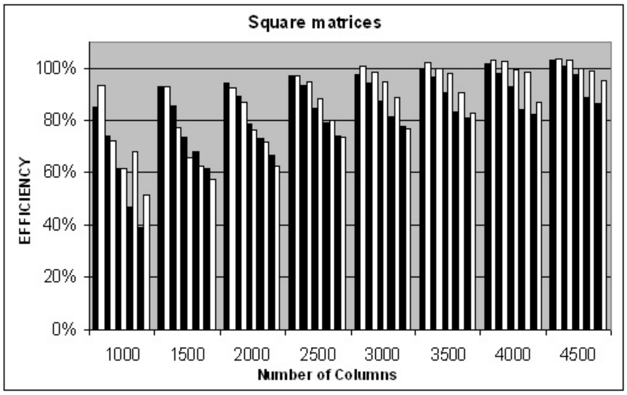


FIG. 4. *PDGEBRD versus modified (efficiency).*

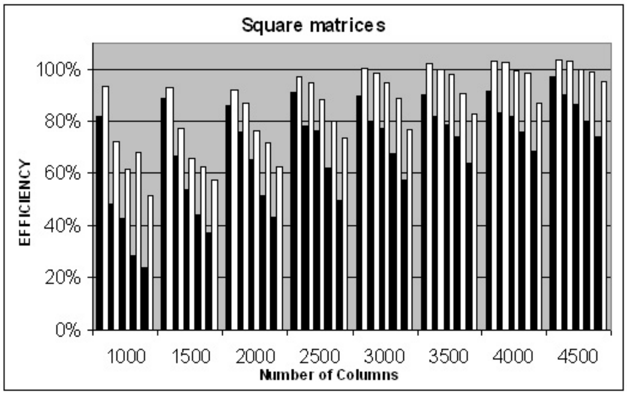


FIG. 5. *PDGEBRD versus modified (efficiency on a linear array of processors).*

products in parallel. This procedure (dubbed GLOBAL.SDOT in [25]) encapsulates all the communication that is required in the parallel algorithm, provided that each processor gets full rows of the matrix. For this reason we do not use a two-dimensional block cyclic distribution. Note that the ScaLAPACK routine PDGEBRD does require communication not only to compute the dot products but also to compute and apply the Householder reflectors. Our one-dimensional distribution, together with the acceptance of some redundancy in the arithmetic, due to the computation of the Householder vectors, produces a parallel algorithm which is well load-balanced and reduces significantly the communication needs, as compared to the ScaLAPACK code. The communication overhead expressed in (12) allows us to conclude (see [25]) that our parallel algorithm is efficient provided that $m/p$ is large enough.

We have described the methodology employed to develop our sequential and parallel codes which intends to use, as much as possible, calls of LAPACK and ScaLAPACK routines, in order to obtain high levels of portability and efficiency.

Our results show that the sequential code for the new method is competitive with the LAPACK routine DGEBRD; although for square matrices we found DGEBRD to be faster. This is not surprising since DGEBRD requires in this case about $\frac{8}{3}n^3$ flops and the new method uses $\frac{1}{3}n^3$ additional flops. Furthermore, DGEBRD has a better ratio of floating point operations to memory references, because it uses level 2 and level 3 BLAS, whereas the new algorithm does not use any level 3 BLAS routine. Even so, without an initial QR decomposition, the new method is faster than DGEBRD if $m$ is much larger than $n$ (in our tests, this happened with $m = 10000$ and $n = 3500$).

Our experimental results on a multiprocessor system do not show as clearly as we expected initially the superiority of the new method. There is a very good reason for this: the ScaLAPACK routine PDGEBRD proved to be very efficient in our tests; this is due to the fact that the communications in our machine are fast and also because we used a maximum of 10 processors only. Since the new algorithm reduces the communication overheads, its virtues will emerge whenever the cost of communication becomes higher comparatively to computation time (a larger number of processors and/or slower communications). Nevertheless, for rectangular matrices our parallel code was marginally faster than PDGEBRD. We expect to be able to use a larger number of processors in the very near future in order to be able to support our claim that the new method has better scalability than the ScaLAPACK routine.

The modification proposed in this paper for the new algorithm reduces to half the number of communication events. In our tests, the gain has not been dramatic, but it may be much more significant on systems with a larger number of processors and/or larger latency in the communications.

To conclude, let us express our view that the new algorithm is very promising for parallel processing. There is still scope to optimize our code and to make it even more competitive with the highly optimized code of the ScaLAPACK routine.

## REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., Software Environ. Tools 9, SIAM, Philadelphia, 1999.

[2] P. Arbenz, *Divide-and-conquer algorithms for the computation of the SVD of bidiagonal matrices*, in Vector and Parallel Computing, Ellis Horwood Ser. Comput. Appl., Horwood, Chichester, UK, 1989, pp. 1–10.

[3] J. Barlow, *More accurate bidiagonal reduction for computing the singular value decomposition*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 761–798.

[4] J. Barlow, N. Bosner, and Z. Drmač, *A new stable bidiagonal reduction algorithm*, Linear Algebra Appl., 397 (2005), pp. 35–84.

[5] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.

[6] A. Björck, *Solving linear least squares problems by Gram-Schmidt orthogonalization*, BIT, 7 (1967), pp. 1–21.

[7] A. Björck, *Numerics of Gram-Schmidt orthogonalization*, Linear Algebra Appl., 197/198 (1994), pp. 297–316.

[8] L. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. Whaley, *ScaLAPACK Users' Guide*, Software Environ. Tools 4, SIAM, Philadelphia, 1997.

[9] N. Bosner and J. Barlow, *Block and Parallel Versions of One-Sided Bidiagonalization*, Tech. report, University of Zagreb, Zagreb, Croatia, 2005; poster available on http://osijek.fernuni-hagen.de/∼luka/Presentations/Nela.pdf.

[10] N. Bosner and Z. Drmač, *On accuracy properties of one-sided bidiagonalization algorithm and its applications*, in Proceedings of the Conference on Applied Mathematics and Scientific Computing, Z. Drmač, M. Marušić, and Z. Tutek, eds., Springer, Dordrecht, 2005, pp. 141–150.

[11] T. F. Chan, *An improved algorithm for computing the singular value decomposition*, ACM Trans. Math. Software, 8 (1982), pp. 72–83.

[12] T. F. Chan, *Rank revealing QR factorizations*, Linear Algebra Appl., 88/89 (1987), pp. 67–82.

[13] J. Choi, J. Dongarra, S. Ostrouchov, A. Petitet, D. Walker, and R. Whaley, *A proposal for a set of parallel basic linear algebra subprograms*, LAPACK Working Note 100, University of Tennessee, Knoxville, TN, 1995.

[14] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[15] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, LAPACK Working Note 119, University of Tennessee, Knoxville, TN, 1997.

[16] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.

[17] J. Dongarra, J. Croz, S. Hammarling, and R. Hanson, *An extended set of FORTRAN basic linear algebra subprograms*, ACM Trans. Math. Software, 14 (1988), pp. 1–17.

[18] K. V. Fernando and B. N. Parlett, *Accurate singular values and differential QD algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[19] G. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[20] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solution*, Numer. Math., 14 (1970), pp. 403–420.

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[22] M. Gu and S. C. Eisenstat, *A divide-and-conquer algorithm for the bidiagonal SVD*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 79–92.

[23] IEEE, *IEEE Standard for Binary Floating Point Arithmetic*, ANSI/IEEE std 754/1985, IEEE Computer Society, Los Alamitos, CA, 1985.

[24] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Englewood Cliffs, NJ, 1980.

[25] R. Ralha, *A new algorithm for singular value decompositions*, in Proceedings of the Second Euromicro Workshop on Parallel and Distributed Processing, IEEE Computer Society, Los Alamitos, CA, 1994, pp. 240–244.

[26] R. Ralha, *One-sided reduction to bidiagonal form*, Linear Algebra Appl., 358 (2003), pp. 219–238.

[27] R. Ralha and A. Mackiewicz, *An efficient algorithm for the computation of singular values*, in Proceedings of the Third International Congress of Numerical Methods in Engineering (Zaragoza, Spain), M. Doblaré, J. M. Correas, E. Alarcón, L. Gavete, and M. Pastor, eds., Spanish Society of Numerical Methods in Engineering, Barcelona, Spain, pp. 1371–1380, 1996.

[28] Scali AS, *Scali System Guide*, http://www.scali.com, 2002.

[29] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*, MIT Press, Cambridge, MA, 1996.

[30] J. H.Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 838–849

# ON NORMWISE STRUCTURED BACKWARD ERRORS FOR SADDLE POINT SYSTEMS*

HUA XIANG† AND YIMIN WEI‡

**Abstract.** We derive the explicit expressions of the normwise structured backward errors of saddle point systems; we extend the previous results of Sun [*Linear Algebra Appl.*, 288 (1999), pp. 75–88] to a general $2 \times 2$ block linear system, where the (1,1) block is general and not symmetric. We also compare the structured backward error with the unstructured one and find that the difference can be arbitrarily large.

**Key words.** saddle point systems, Karush–Kuhn–Tucker (KKT) systems, backward error

**AMS subject classifications.** 15A06, 65F99, 65G99

**DOI.** 10.1137/060663684

## 1. Introduction.

Backward error is of great importance in numerical analysis. It can answer how close the problem that is actually solved is to the one we want to solve and reveals the stability of a numerical method [12]. For solving a general linear system, there exist explicit expressions for the normwise and componentwise backward errors [15, 16]. Here we consider a special kind of linear system—the saddle point system—as follows:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times m}$. When the (1,1) block $A$ is symmetric, (1.1) is called a Karush–Kuhn–Tucker (KKT) system. In addition, (1.1) is also called an equilibrium equation [19, 22] or an augmented system [2, 11]. Saddle point systems occur in many areas of computational science and engineering (see [1] and the references therein) such as computational fluid dynamics (CFD) [6, 7, 8, 14], electrical circuits and networks [18, 19], and constrained optimization [25, 26], constrained and weighted least squares problems [5, 10, 21, 24].

For simplicity, we rewrite (1.1) as

$$\mathcal{A}z = d. \tag{1.2}$$

Assuming that the computed solution is $\widetilde{z} = [\widetilde{u}^T, \widetilde{p}^T]^T$, we define the normwise unstructured backward error $\eta(\widetilde{z})$ as

$$\eta(\widetilde{z}) := \min_{\Delta\mathcal{A},\Delta d} \left\{ \left\| \begin{bmatrix} \frac{\|\Delta\mathcal{A}\|_F}{\|\mathcal{A}\|_F}, & \frac{\|\Delta d\|_2}{\|d\|_2} \end{bmatrix} \right\|_2 : (\mathcal{A} + \Delta\mathcal{A})\widetilde{z} = d + \Delta d \right\},$$

---

[*] Received by the editors June 24, 2006; accepted for publication (in revised form) by A. J. Wathen February 2, 2007; published electronically July 25, 2007. This project is supported by the National Natural Science Foundation of China under grant 10471027 and the Shanghai Education Committee.
http://www.siam.org/journals/simax/29-3/66368.html

[†] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, People's Republic of China. Current address: INRIA Futurs, Parc Club Orsay Université, 4 rue Jacques Monod - Bât G, 91893 Orsay Cedex, France (hua.xiang@inria.fr).

[‡] Corresponding author. School of Mathematical Sciences, Fudan University, Shanghai, 200433, People's Republic of China and Key Laboratory of Mathematics for Nonlinear Sciences, Fudan University, Ministry of Education, Shanghai, 200433, People's Republic of China (ymwei@fudan.edu.cn).

which can be expressed as

$$\eta(\widetilde{z}) = \frac{\|d - \mathcal{A}\widetilde{z}\|_2}{\sqrt{\|\mathcal{A}\|_F^2 \|\widetilde{z}\|_2^2 + \|d\|_2^2}}. \tag{1.3}$$

We have discussed the structured condition numbers of (1.1) [28]. Here we will consider its structured backward error. Taking into consideration the special block structure of (1.1), we define the normwise structured backward error as

$$\eta_S(\widetilde{u}, \widetilde{p}) := \min_{\{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} \in\ \mathcal{F}} \left\| \left[ \tfrac{\|\Delta A\|_F}{\|A\|_F}, \tfrac{\|\Delta B\|_F}{\|B\|_F}, \tfrac{\|\Delta f\|_2}{\|f\|_2}, \tfrac{\|\Delta g\|_2}{\|g\|_2} \right] \right\|_2,$$

where $\mathcal{F}$ is defined by

$$\mathcal{F} = \left\{ \{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} : \begin{bmatrix} A + \Delta A & (B + \Delta B)^T \\ B + \Delta B & 0 \end{bmatrix} \begin{bmatrix} \widetilde{u} \\ \widetilde{p} \end{bmatrix} = \begin{bmatrix} f + \Delta f \\ g + \Delta g \end{bmatrix} \right\}.$$

Following the definition in [20], we further define

$$\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u}, \widetilde{p}) := \min_{\{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} \in \mathcal{F}} \left\| \left[ \|\Delta A\|_F, \theta\|\Delta B\|_F, \lambda\|\Delta f\|_2, \mu\|\Delta g\|_2 \right] \right\|_2.$$

If $A \neq 0$, $B \neq 0$, $f \neq 0$, and $g \neq 0$, then taking $\theta_* = \|A\|_F / \|B\|_F$, $\lambda_* = \|A\|_F / \|f\|_2$, $\mu_* = \|A\|_F / \|g\|_2$, we have $\eta_S(\widetilde{u}, \widetilde{p}) = \frac{1}{\|A\|_F} \eta^{(\theta_*,\ \lambda_*,\ \mu_*)}(\widetilde{u}, \widetilde{p})$.

If $\eta(\widetilde{z})$ is small, then the computed solution $\widetilde{z}$ satisfies a nearby system $(\mathcal{A} + \Delta\mathcal{A})\widetilde{z} = d + \Delta d$, where $\|\Delta A\|_F$ and $\|\Delta d\|_2$ are relatively small. We can say that the algorithm is normwise backward stable. But the perturbed coefficient $\mathcal{A} + \Delta\mathcal{A}$ may not have the saddle point form like (1.1). If we require the perturbation $\Delta\mathcal{A}$ to preserve its original structure, $\|\Delta\mathcal{A}\|_F$ may not be small anymore; that is, the structured backward error $\eta_S(\widetilde{u}, \widetilde{p})$ may be large. However, if $\eta_S(\widetilde{u}, \widetilde{p})$ is small, then the algorithm is strongly stable [3, 4], and we solve a nearby saddle point system. A stable algorithm for solving saddle point systems is not necessarily a strongly stable one [20].

In many cases, there is no perturbation in the (1,1) block; for example, in linear least squares problems (LLSPs), the (1,1) block $A$ is the identity matrix $I$ and is not allowed to be perturbed. So we define

$$\gamma_S(\widetilde{u}, \widetilde{p}) := \min_{\{\Delta B,\ \Delta f,\ \Delta g\} \in\ \mathcal{G}} \left\| \left[ \tfrac{\|\Delta B\|_F}{\|B\|_F}, \tfrac{\|\Delta f\|_2}{\|f\|_2}, \tfrac{\|\Delta g\|_2}{\|g\|_2} \right] \right\|_2,$$

where $\mathcal{G}$ is determined by

$$\mathcal{G} = \left\{ \{\Delta B,\ \Delta f,\ \Delta g\} : \begin{bmatrix} A & (B + \Delta B)^T \\ B + \Delta B & 0 \end{bmatrix} \begin{bmatrix} \widetilde{u} \\ \widetilde{p} \end{bmatrix} = \begin{bmatrix} f + \Delta f \\ g + \Delta g \end{bmatrix} \right\}.$$

To derive $\gamma_S(\widetilde{u}, \widetilde{p})$, we need

$$\gamma^{(\lambda,\mu)}(\widetilde{u}, \widetilde{p}) := \min_{\{\Delta B,\ \Delta f,\ \Delta g\} \in\ \mathcal{G}} \left\| \left[ \|\Delta B\|_F, \lambda\|\Delta f\|_2, \mu\|\Delta g\|_2 \right] \right\|_2.$$

We can verify that $\gamma_S(\widetilde{u}, \widetilde{p}) = \frac{1}{\|B\|_F} \gamma^{(\lambda^*,\mu^*)}(\widetilde{u}, \widetilde{p})$, where $\lambda^* = \frac{\|B\|_F}{\|f\|_2}$, $\mu^* = \frac{\|B\|_F}{\|g\|_2}$.

Let us review some previous results. In [20] Sun investigated the case where the (1,1) block $A$ is symmetric and defined

$$\eta_{\mathrm{sym}}(\widetilde{u}, \widetilde{p}) := \min_{\{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} \in\ \mathcal{E}} \left\| \left[ \tfrac{\|\Delta A\|_F}{\|A\|_F}, \tfrac{\|\Delta B\|_F}{\|B\|_F}, \tfrac{\|\Delta f\|_2}{\|f\|_2}, \tfrac{\|\Delta g\|_2}{\|g\|_2} \right] \right\|_2,$$

where

$$\mathcal{E} = \left\{\{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} : \{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} \in \mathcal{F}, (\Delta A)^T = \Delta A\right\}.$$

In the deduction, the following definition is also needed:

$$\eta_{\text{sym}}^{(\theta,\ \lambda,\ \mu)}(\widetilde{u}, \widetilde{p}) := \min_{\{\Delta A,\ \Delta B,\ \Delta f,\ \Delta g\} \in \mathcal{E}} \left\| \left[\|\Delta A\|_F, \theta\|\Delta B\|_F, \lambda\|\Delta f\|_2, \mu\|\Delta g\|_2\right] \right\|_2.$$

A special case, where $A = I$ and $g = 0$, is considered in [13]. In that case, there are no perturbations in $A$ and $g$. In this paper, we first consider the case where $\Delta A \neq 0$ and the (1,1) block $A$ is a general matrix, which is the case arising from CFD. We then investigate the case where no perturbation in the (1,1) block is permitted; i.e., $\Delta A = 0$. We categorize the problem into two classes because we should treat these two cases with different methods. We give the explicit expressions for the normwise structured backward errors of these two cases. The special case $\Delta B = 0$, $\Delta f = 0$, or $\Delta g = 0$ can be derived directly.

Before our discussion, we need the following three lemmas, which can be found in [20].

LEMMA 1. [illegible] $f \neq 0 \in \mathbb{R}^m$ [illegible] $g \in \mathbb{R}^n$ [illegible] $Xf = g$ [illegible]

$$X^T = f^{\dagger^T} g^T + (I - ff^\dagger)Z,$$

[illegible] $Z \in \mathbb{R}^{m \times n}$ [illegible] $f^\dagger$ [illegible] [23].

LEMMA 2. [illegible] $F \in \mathbb{R}^{p \times m}, G \in \mathbb{R}^{n \times q}$, [illegible] $K \in \mathbb{R}^{p \times q}$ [illegible] $X_* = F^\dagger K G^\dagger$. [illegible]

$$\min_{X \in \mathbb{R}^{m \times n}} \|FXG - K\|_F = \|FX_*G - K\|_F.$$

LEMMA 3. [illegible] $\Phi$ [illegible]

$$\Phi = \begin{bmatrix} \alpha I_m + \beta\xi uu^T & -\xi up^T \\ -\xi pu^T & \gamma I_n + \xi pp^T \end{bmatrix},$$

[illegible] $u \in \mathbb{R}^m, p \in \mathbb{R}^n$ $\alpha, \beta, \gamma,$ [illegible] $\xi$ [illegible] $\alpha, \beta, \gamma > 0$ [illegible]

$$\Phi^{-1} = \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{12}^T & \Psi_{22} \end{bmatrix}, \quad \Psi_{11} \in \mathbb{R}^{m \times m}.$$

[illegible]

$$\Psi_{11} = \frac{1}{\alpha}\Big(I_m - \xi(\beta\gamma + \beta\gamma\|p\|_2^2 - \xi\|p\|_2^2)uu^T/\psi\Big),$$

$$\Psi_{22} = \frac{1}{\gamma}\Big(I_n - \xi(\alpha + \beta\xi\|u\|_2^2 - \xi\|u\|_2^2)pp^T/\psi\Big), \quad \Psi_{12} = \xi up^T/\psi,$$

[illegible] $\psi = \alpha(\gamma + \xi\|p\|_2^2) + \xi\|u\|_2^2(\beta\gamma + \beta\xi\|p\|_2^2 - \xi\|p\|_2^2)$

The rest of this paper is organized as follows. In section 2, we derive the explicit expression of $\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u}, \widetilde{p})$, which yields the normwise structured backward error. In section 3, we give an upper bound of $\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u}, \widetilde{p})$ and show that the structured backward error can be arbitrarily larger than the normal unstructured one. Then numerical examples are illustrated in section 4.

**2. Expression of $\boldsymbol{\eta^{(\theta,\ \lambda,\ \mu)}(\tilde{u},\tilde{p})}$.** Let $[\widetilde{u}^T,\widetilde{p}^T]^T$ be the computed solution, and let $\widetilde{u}\neq 0$. To derive the expression of $\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u},\widetilde{p})$, we need $\eta^{(\theta)}(\widetilde{u},\widetilde{p})$ first, which is defined as

$$\eta^{(\theta)}(\widetilde{u},\widetilde{p}) := \min\left\{ \big\| \left[ \|\Delta A\|_F, \theta\|\Delta B\|_F \right] \big\|_2 : \begin{bmatrix} A+\Delta A & (B+\Delta B)^T \\ B+\Delta B & 0 \end{bmatrix} \begin{bmatrix} \widetilde{u} \\ \widetilde{p} \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \right\}.$$

We now consider only the perturbation of the coefficient matrix, such that

$$\begin{bmatrix} A+\Delta A & (B+\Delta B)^T \\ B+\Delta B & 0 \end{bmatrix} \begin{bmatrix} \widetilde{u} \\ \widetilde{p} \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \tag{2.1}$$

Let $r_f := f - A\widetilde{u} - B^T\widetilde{p}$, $r_g := g - B\widetilde{u}$ be the computed residuals. In the following $\|\cdot\|$ stands for the 2-norm, where we omit the subscript for simplicity. The following lemma gives the explicit expression of $\eta^{(\theta)}(\widetilde{u},\widetilde{p})$.

LEMMA 4. [illegible] $\eta^{(\theta)}(\widetilde{u},\widetilde{p})$ [illegible]

$$\left[\eta^{(\theta)}(\widetilde{u},\widetilde{p})\right]^2 = \frac{\theta^2\|r_g\|^2}{\|\widetilde{u}\|^2} + \frac{(r_f^T\widetilde{u} - r_g^T\widetilde{p})^2}{\|\widetilde{u}\|^4} + \frac{\theta^2[\|\widetilde{u}\|^2\|r_f\|^2 - (r_f^T\widetilde{u})^2]}{\|\widetilde{u}\|^2(\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2)}. \tag{2.2}$$

[illegible]. According to the second formula of (2.1), we have

$$\Delta B\ \widetilde{u} = r_g.$$

Using Lemma 1, we get

$$(\Delta B)^T = \widetilde{u}^{\dagger^T} r_g^T + (I - \widetilde{u}\widetilde{u}^\dagger)Z,$$

where $Z \in \mathbb{R}^{m\times n}$, $\widetilde{u}^\dagger$ denotes the Moore–Penrose inverse of $\widetilde{u}$. Substituting it into the first formula of (2.1), we obtain

$$\Delta A\ \widetilde{u} = r_f - (r_g^T\widetilde{p})\widetilde{u}^{\dagger^T} - (I - \widetilde{u}\widetilde{u}^\dagger)Z\widetilde{p}.$$

Again applying Lemma 1, we deduce that

$$\Delta A^T = \widetilde{u}^{\dagger^T}[r_f - (r_g^T\widetilde{p})\widetilde{u}^{\dagger^T} - (I - \widetilde{u}\widetilde{u}^\dagger)Z\widetilde{p}\ ]^T + (I - \widetilde{u}\widetilde{u}^\dagger)W,$$

where $W \in \mathbb{R}^{m\times m}$. With the definition $F := (I - \widetilde{u}\widetilde{u}^\dagger)Z$, $G := r_f\widetilde{u}^\dagger - (r_g^T\widetilde{p})\widetilde{u}^{\dagger^T}\widetilde{u}^\dagger$, we have

$$\begin{aligned}
\|\Delta A\|_F^2 + \theta^2\|\Delta B\|_F^2 &= \theta^2\|r_g\widetilde{u}^\dagger\|_F^2 + \theta^2\|F\|_F^2 + \|G^T - \widetilde{u}^{\dagger^T}\widetilde{p}^T F^T\|_F^2 + \|(I - \widetilde{u}\widetilde{u}^\dagger)W\|_F^2 \\
&= \theta^2\mathrm{tr}(FF^T) + \mathrm{tr}\Big((G - F\widetilde{p}\widetilde{u}^\dagger)(G^T - \widetilde{u}^{\dagger^T}\widetilde{p}^T F^T)\Big) + \theta^2\|r_g\widetilde{u}^\dagger\|_F^2 + \|(I - \widetilde{u}\widetilde{u}^\dagger)W\|_F^2 \\
&= \mathrm{tr}\Big(F[\theta^2 I + \widetilde{p}\widetilde{u}^\dagger(\widetilde{p}\widetilde{u}^\dagger)^T]F^T - F(\widetilde{p}\widetilde{u}^\dagger G^T) - (\widetilde{p}\widetilde{u}^\dagger G^T)^T F^T\Big) \\
&\quad + \theta^2\|r_g\widetilde{u}^\dagger\|_F^2 + \|G\|_F^2 + \|(I - \widetilde{u}\widetilde{u}^\dagger)W\|_F^2.
\end{aligned}$$

The formula above can be rewritten as

$$\|\Delta A\|_F^2 + \theta^2\|\Delta B\|_F^2 = \|M - N\|_F^2 - \|N\|_F^2 + \theta^2\|r_g\widetilde{u}^\dagger\|_F^2 + \|G\|_F^2 + \|(I - \widetilde{u}\widetilde{u}^\dagger)W\|_F^2,$$

where $M := F[\theta^2 I + \widetilde{p}\widetilde{u}^\dagger(\widetilde{p}\widetilde{u}^\dagger)^T]^{\frac{1}{2}}, N^T := [\theta^2 I + \widetilde{p}\widetilde{u}^\dagger(\widetilde{p}\widetilde{u}^\dagger)^T]^{-\frac{1}{2}}\widetilde{p}\widetilde{u}^\dagger G^T$.

Applying Lemma 2, we can obtain

$$\min_Z \|M - N\|_F^2 = \min_Z \left\|(I - \widetilde{u}\widetilde{u}^\dagger)Z[\theta^2 I + \widetilde{p}\widetilde{u}^\dagger(\widetilde{p}\widetilde{u}^\dagger)^T]^{\frac{1}{2}} - N\right\|_F^2 = \|\widetilde{u}\widetilde{u}^\dagger N\|_F^2,$$

and the minimum is obtained at $Z_{\min} = -(I - \widetilde{u}\widetilde{u}^\dagger)^\dagger N[\theta^2 I + \widetilde{p}\widetilde{u}^\dagger(\widetilde{p}\widetilde{u}^\dagger)^T]^{-\frac{1}{2}}$. Noticing that $\widetilde{u}\widetilde{u}^\dagger$, $I - \widetilde{u}\widetilde{u}^\dagger$ are orthogonal projectors, we get $\|N\|_F^2 - \|\widetilde{u}\widetilde{u}^\dagger N\|_F^2 = \operatorname{tr}(N^T(I - \widetilde{u}\widetilde{u}^\dagger)N) = \|(I - \widetilde{u}\widetilde{u}^\dagger)N\|_F^2$, and so

$$\left[\eta^{(\theta)}(\widetilde{u},\widetilde{p})\right]^2 = \min_{Z,W}\left\{\|\Delta A\|_F^2 + \theta^2\|\Delta B\|_F^2\right\} = \theta^2\|r_g\|^2\|\widetilde{u}^\dagger\|^2 + \|G\|_F^2 - \|(I - \widetilde{u}\widetilde{u}^\dagger)N\|_F^2.$$

Therefore,

$$\begin{aligned}\left[\eta^{(\theta)}(\widetilde{u},\widetilde{p})\right]^2 &= \frac{\theta^2\|r_g\|^2}{\|\widetilde{u}\|^2} + \frac{1}{\|\widetilde{u}\|^4}\left\|r_f\widetilde{u}^T - \frac{r_g^T\widetilde{p}}{\|\widetilde{u}\|^2}\widetilde{u}\widetilde{u}^T\right\|_F^2 \\ &\quad - \frac{1}{\|\widetilde{u}\|^4}\left\|\left(\theta^2 I + \frac{\widetilde{p}\widetilde{p}^T}{\|\widetilde{u}\|^2}\right)^{-\frac{1}{2}}\left(\widetilde{p}r_f^T - \frac{r_f^T\widetilde{u}}{\|\widetilde{u}\|^2}\widetilde{p}\widetilde{u}^T\right)\right\|_F^2. \qquad (2.3)\end{aligned}$$

The formula above can be simplified. The second term of (2.3) is equal to

$$\begin{aligned}&\frac{1}{\|\widetilde{u}\|^4}\operatorname{tr}\left[\left(\widetilde{u}r_f^T - \frac{r_g^T\widetilde{p}}{\|\widetilde{u}\|^2}\widetilde{u}\widetilde{u}^T\right)\left(r_f\widetilde{u}^T - \frac{r_g^T\widetilde{p}}{\|\widetilde{u}\|^2}\widetilde{u}\widetilde{u}^T\right)\right] \\ &= \frac{1}{\|\widetilde{u}\|^4}\left[\|\widetilde{u}\|^2\|r_f\|^2 - 2(r_f^T\widetilde{u})(r_g^T\widetilde{p}) + (r_g^T\widetilde{p})^2\right].\end{aligned}$$

And the third term of (2.3) is equivalent to

$$\begin{aligned}&-\frac{1}{\|\widetilde{u}\|^4}\operatorname{tr}\left[\left(r_f\widetilde{p}^T - \frac{r_f^T\widetilde{u}}{\|\widetilde{u}\|^2}\widetilde{u}\widetilde{p}^T\right)\left(\theta^2 I + \frac{\widetilde{p}\widetilde{p}^T}{\|\widetilde{u}\|^2}\right)^{-1}\left(\widetilde{p}r_f^T - \frac{r_f^T\widetilde{u}}{\|\widetilde{u}\|^2}\widetilde{p}\widetilde{u}^T\right)\right] \\ &= -\frac{\|\widetilde{p}\|^2}{\|\widetilde{u}\|^2(\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2)}\left[\|r_f\|^2 - \frac{(r_f^T\widetilde{u})^2}{\|\widetilde{u}\|^2}\right],\end{aligned}$$

where we use the Sherman–Morrison–Woodbury formula [9]

$$\left(\theta^2 I + \frac{\widetilde{p}\widetilde{p}^T}{\|\widetilde{u}\|^2}\right)^{-1} = \theta^{-2}\left(I - \frac{\widetilde{p}\widetilde{p}^T}{\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2)}\right).$$

Hence, $\eta^{(\theta)}(\widetilde{u},\widetilde{p})$ can be expressed as

$$\begin{aligned}\left[\eta^{(\theta)}(\widetilde{u},\widetilde{p})\right]^2 &= \frac{\theta^2\|r_g\|^2}{\|\widetilde{u}\|^2} + \frac{(r_g^T\widetilde{p})^2}{\|\widetilde{u}\|^4} - \frac{2(r_f^T\widetilde{u})(r_g^T\widetilde{p})}{\|\widetilde{u}\|^4} + \frac{\theta^2\|r_f\|^2}{\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2} \\ &\quad + \frac{\|\widetilde{p}\|^2(r_f^T\widetilde{u})^2}{\|\widetilde{u}\|^4(\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2)},\end{aligned}$$

which is equivalent with the expression (2.2). $\square$

In the case where (1,1) block $A$ is symmetric, we define

$$\eta^{(\theta)}_{\text{sym}}(\widetilde{u},\widetilde{p}) := \min_{\{\Delta A,\ \Delta B,\ 0,\ 0\}\in\mathcal{E}} \left\| \left[\|\Delta A\|_F, \theta\|\Delta B\|_F\right] \right\|_2 .$$

Its expression is given by Sun [20]:

$$\left[\eta^{(\theta)}_{\text{sym}}(\widetilde{u},\widetilde{p})\right]^2 = \frac{\theta^2\|r_g\|^2}{\|\widetilde{u}\|^2} + \frac{(r_f^T\widetilde{u} - r_g^T\widetilde{p})^2}{\|\widetilde{u}\|^4} + \frac{2\theta^2[\|\widetilde{u}\|^2\|r_f\|^2 - (r_f^T\widetilde{u})^2]}{\|\widetilde{u}\|^2(\theta^2\|\widetilde{u}\|^2 + 2\|\widetilde{p}\|^2)}.$$

For convenience, we denote $\eta^{(\theta)}(\widetilde{u},\widetilde{p})$ as

$$\left[\eta^{(\theta)}(\widetilde{u},\widetilde{p})\right]^2 = \theta^2\tau\|r_g\|^2 + (\tau-\sigma)\|r_f\|^2 + \tau^2(r_g^T\widetilde{p})^2 - 2\tau^2(r_f^T\widetilde{u})(r_g^T\widetilde{p}) + \tau\sigma(r_f^T\widetilde{u})^2,$$

where $\tau := \|\widetilde{u}\|^{-2}$, $\sigma := \tau^2\|\widetilde{p}\|^2(\theta^2 + \tau\|p\|^2)^{-1}$. In addition we define $\rho := \tau - \sigma + \lambda^2$.

We now consider perturbations both in the coefficient matrix and the right-hand side:

$$\begin{bmatrix} A+\Delta A & (B+\Delta B)^T \\ B+\Delta B & 0 \end{bmatrix} \begin{bmatrix} \widetilde{u} \\ \widetilde{p} \end{bmatrix} = \begin{bmatrix} f+\Delta f \\ g+\Delta g \end{bmatrix}, \tag{2.4}$$

where $\widetilde{u} \neq 0$. We use $\eta^{(\theta)}(\widetilde{u},\widetilde{p})$ to derive the expression of $\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u},\widetilde{p})$, which is given in the following theorem.

THEOREM 1. [illegible]

$$\begin{aligned}[\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u},\widetilde{p})]^2 = {} & \frac{\theta^2\mu^2}{\Omega_0}\|r_g\|^2 + \frac{\lambda^2\mu^4\|\widetilde{u}\|^2}{\Omega_0\Omega}(r_g^T\widetilde{p})^2 - \frac{2\lambda^2\mu^2}{\Omega}(r_f^T\widetilde{u})(r_g^T\widetilde{p}) \\ & + \frac{\theta^2\lambda^2}{\Omega_1}\|r_f\|^2 + \frac{\lambda^4\mu^2\|\widetilde{p}\|^2}{\Omega_1\Omega}(r_f^T\widetilde{u})^2, \end{aligned} \tag{2.5}$$

[illegible]

$$\begin{aligned} \Omega_0 &:= \theta^2 + \mu^2\|\widetilde{u}\|^2, \\ \Omega_1 &:= \theta^2 + \lambda^2\|\widetilde{p}\|^2 + \theta^2\lambda^2\|\widetilde{u}\|^2, \\ \Omega\ &:= \theta^2 + \mu^2\|\widetilde{u}\|^2 + \lambda^2\|\widetilde{p}\|^2 + \theta^2\lambda^2\|\widetilde{u}\|^2 + \lambda^2\mu^2\|\widetilde{u}\|^4. \end{aligned}$$

[illegible]. By the definition, we have

$$\begin{aligned}[\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u},\widetilde{p})]^2 &= \min_{\Delta f,\ \Delta g}\left\{\lambda^2\|\Delta f\|^2 + \mu^2\|\Delta g\|^2 + \min_{\Delta A,\ \Delta B}\|[\|\Delta A\|_F, \theta\|\Delta B\|_F]\|_F^2\right\} \\ &:= \min_{\Delta f,\ \Delta g}\chi(\Delta f, \Delta g), \end{aligned}$$

where

$$\begin{aligned}\chi(\Delta f,\Delta g) &= \lambda^2\|\Delta f\|^2 + \mu^2\|\Delta g\|^2 + \theta^2\tau\|r_g + \Delta g\|^2 + (\tau-\sigma)\|r_f + \Delta f\|^2 \\ &\quad + \tau^2[(r_g+\Delta f)^T\widetilde{p}]^2 - 2\tau^2[(r_f+\Delta f)^T\widetilde{u}][(r_g+\Delta g)^T\widetilde{p}] \\ &\quad + \tau\sigma[(r_f+\Delta f)^T\widetilde{u}]^2 \\ &= [\eta^{(\theta)}(\widetilde{u},\widetilde{p})]^2 + 2\begin{bmatrix}\Delta f \\ \Delta g\end{bmatrix}^T \begin{bmatrix}(\tau-\sigma)r_f + (\tau\sigma r_f^T\widetilde{u} - \tau^2 r_g^T\widetilde{p})\widetilde{u} \\ \tau\theta^2 r_g + \tau^2(r_g^T\widetilde{p} - r_f^T\widetilde{u})\widetilde{p}\end{bmatrix} \\ &\quad + \begin{bmatrix}\Delta f \\ \Delta g\end{bmatrix}^T \begin{bmatrix}\rho I_m + \tau\sigma\widetilde{u}\widetilde{u}^T & -\tau^2\widetilde{u}\widetilde{p}^T \\ -\tau^2\widetilde{p}\widetilde{u}^T & (\tau\theta^2+\mu^2)I_n + \tau^2\widetilde{p}\widetilde{p}^T\end{bmatrix}\begin{bmatrix}\Delta f \\ \Delta g\end{bmatrix}.\end{aligned}$$

We denote the formula above as

$$\chi(w) = [\eta^{(\theta)}(\widetilde{u}, \widetilde{p})]^2 + 2w^T b + w^T \Phi w.$$

Obviously, the minimum is obtained when $w = -\Phi^{-1} b$. Correspondingly,

$$[\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u}, \widetilde{p})]^2 = [\eta^{(\theta)}(\widetilde{u}, \widetilde{p})]^2 - b^T \Phi^{-1} b. \tag{2.6}$$

Applying Lemma 3, we have

$$\Phi^{-1} = \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{12}^T & \Psi_{22} \end{bmatrix},$$

and

$$\begin{aligned} \Psi_{11} &= \frac{1}{\rho}\left(I_m - \frac{\tau\xi}{\psi}\widetilde{u}\widetilde{u}^T\right), \\ \Psi_{22} &= \frac{1}{\tau\theta^2 + \mu^2}\left(I_n - \frac{\tau^2\lambda^2}{\psi}\widetilde{p}\widetilde{p}^T\right), \\ \Psi_{12} &= \frac{\tau^2}{\psi}\widetilde{u}\widetilde{p}^T, \end{aligned}$$

where $\xi := \sigma(\tau\theta^2 + \mu^2) + \tau^2(\sigma - \tau)\|\widetilde{p}\|^2$, and $\psi := \rho(\tau\theta^2 + \mu^2 + \tau^2\|\widetilde{p}\|^2) + \xi$. After tedious computation, we obtain

$$b^T \Phi^{-1} b = K_{0f}\|r_f\|^2 + K_{0g}\|r_g\|^2 + K_{1f}(r_f^T \widetilde{u})^2 + K_{1g}(r_g^T \widetilde{p})^2 + K_{fg}(r_f^T \widetilde{u})(r_g^T \widetilde{p}),$$

where

$$\begin{aligned} K_{0f} &:= \frac{\theta^4}{(\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2)\Omega_1}, \\ K_{0g} &:= \frac{\theta^4}{\|\widetilde{u}\|^2\Omega_0}, \\ K_{1f} &:= \frac{\|\widetilde{p}\|^2}{\|\widetilde{u}\|^4(\theta^2\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2)} - \frac{\lambda^4\mu^2\|\widetilde{p}\|^2}{\Omega_1\Omega}, \\ K_{1g} &:= \frac{1}{\|\widetilde{u}\|^4} - \frac{\lambda^2\mu^4\|\widetilde{u}\|^2}{\Omega_0\Omega}, \\ K_{fg} &:= -\frac{2}{\|\widetilde{u}\|^4} + \frac{2\lambda^2\mu^2}{\Omega}. \end{aligned}$$

Combining this with the expression of $\eta^{(\theta)}(\widetilde{u}, \widetilde{p})$, we obtain (2.5). ∎

[illegible] 1. If we let $\sigma = (2\tau^2\|\widetilde{p}\|^2 - \tau\theta^2)(\theta^2 + 2\tau\|p\|^2)^{-1}$ in the former deduction, we can get

$$\begin{aligned} [\eta_{\text{sym}}^{(\theta,\ \lambda,\ \mu)}(\widetilde{u}, \widetilde{p})]^2 = {} & \frac{\theta^2\mu^2}{\Omega_0}\|r_g\|^2 + \frac{\lambda^2\mu^4\|\widetilde{u}\|^2}{\Omega_0\Omega}(r_g^T \widetilde{p})^2 - \frac{2\lambda^2\mu^2}{\Omega}(r_f^T \widetilde{u})(r_g^T \widetilde{p}) \\ & + \frac{2\theta^2\lambda^2}{\Omega_2}\|r_f\|^2 + \frac{2\lambda^4\mu^2\|\widetilde{p}\|^2 - \theta^2\lambda^4\Omega_0}{\Omega_2\Omega}(r_f^T \widetilde{u})^2, \end{aligned} \tag{2.7}$$

where $\Omega_2 := 2\theta^2 + 2\lambda^2\|\tilde{p}\|^2 + \theta^2\lambda^2\|\tilde{u}\|^2 = 2\Omega_1 - \theta^2\lambda^2\|\tilde{u}\|^2$. We can show that (2.7) is equivalent to Sun's result [20].

2. The induction above is taken under the assumption that $\tilde{u} \neq 0$. In the case $\tilde{u} = 0$, we can also derive a similar result:

$$\eta^{(\theta,\ \lambda,\ \mu)}(0, \tilde{p}) = \sqrt{\frac{\theta^2\lambda^2\|r_f\|^2}{\theta^2 + \lambda^2\|\tilde{p}\|^2} + \mu^2\|r_g\|^2}.$$

This is just a special case of (2.5).

3. In many cases where $g = 0$, there should be no perturbation in $g$, i.e., $\Delta g = 0$. For this case, we define

$$\eta^{(\theta,\ \lambda)}(\tilde{u}, \tilde{p}) := \min_{\{\Delta A,\ \Delta B,\ \Delta f\} \in \mathcal{F}_g} \left\| \left[ \|\Delta A\|_F, \theta\|\Delta B\|_F, \lambda\|\Delta f\| \right] \right\|,$$

where

$$\mathcal{F}_g = \left\{ \{\Delta A,\ \Delta B,\ \Delta f\} : \begin{bmatrix} A + \Delta A & (B + \Delta B)^T \\ B + \Delta B & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} \\ \tilde{p} \end{bmatrix} = \begin{bmatrix} f + \Delta f \\ g \end{bmatrix} \right\}.$$

Let $\mu$ tend to $\infty$ in (2.5); we then obtain

$$\begin{aligned} [\eta^{(\theta,\ \lambda)}(\tilde{u}, \tilde{p})]^2 = {} & \frac{\theta^2\lambda^2\|r_f\|^2}{\Omega_1} + \frac{\theta^2\|r_g\|^2}{\|\tilde{u}\|^2} + \frac{\lambda^4\|\tilde{p}\|^2(r_f^T\tilde{u})^2}{\|\tilde{u}\|^2(1 + \lambda^2\|\tilde{u}\|^2)\Omega_1} \\ & + \frac{\lambda^2(r_g^T\tilde{p})^2}{\|\tilde{u}\|^2(1 + \lambda^2\|\tilde{u}\|^2)} - \frac{2\lambda^2(r_f^T\tilde{u})(r_g^T\tilde{p})}{\|\tilde{u}\|^2(1 + \lambda^2\|\tilde{u}\|^2)}. \end{aligned} \tag{2.8}$$

We can also obtain (2.8) the same way we derive $\eta^{\theta,\lambda,\mu}(\tilde{u}, \tilde{p})$. Here we provide a simple and efficient way. Similarly, we can treat the case of $\Delta f = 0$, or $\Delta B = 0$.

4. In the case where the (1,1) block $A$ is not perturbed, we can derive $\gamma^{(\lambda,\ \mu)}(\tilde{u}, \tilde{p})$ [27], which is given by

$$\begin{aligned} [\gamma^{(\lambda,\ \mu)}(\tilde{u}, \tilde{p})]^2 = {} & \frac{\lambda^2}{\Theta_\lambda}\|r_f\|^2 + \frac{\mu^2}{\Theta_\mu}\|r_g\|^2 - \frac{2\lambda^2\mu^2}{\Theta}(r_f^T\tilde{u})(r_g^T\tilde{p}) \\ & + \frac{\lambda^2\mu^2(\Theta_\lambda - 1)}{\Theta_\lambda\Theta}(r_f^T\tilde{u})^2 + \frac{\lambda^2\mu^2(\Theta_\mu - 1)}{\Theta_\mu\Theta}(r_g^T\tilde{p})^2, \end{aligned} \tag{2.9}$$

where $\Theta_\lambda := 1 + \lambda^2\|\tilde{p}\|^2$, $\Theta_\mu := 1 + \mu^2\|\tilde{u}\|^2$ , and $\Theta := 1 + \mu^2\|\tilde{u}\|^2 + \lambda^2\|\tilde{p}\|^2$.

For the case $\Delta g = 0$, we can let $\mu$ tend to $\infty$ in (2.9) and then obtain the result of [13]. We can deal with the case $\Delta f = 0$ or $\Delta B = 0$ similarly.

**3. Comparison between $\eta_S(\tilde{u}, \tilde{p})$ and $\eta(\tilde{z})$.** From (2.4), we have $\Delta\tilde{u} + (\Delta B)^T\tilde{p} - \Delta f = r_f$, $\Delta B\tilde{u} - \Delta g = r_g$. That is,

$$\left[\Delta A,\quad \beta(\Delta B)^T,\quad \lambda\Delta f\right] \begin{bmatrix} \tilde{u} \\ \frac{1}{\beta}\tilde{p} \\ -\frac{1}{\lambda} \end{bmatrix} = r_f, \quad \left[\alpha\Delta B,\quad \mu\Delta g\right] \begin{bmatrix} \frac{1}{\alpha}\tilde{u} \\ -\frac{1}{\mu} \end{bmatrix} = r_g.$$

Using Lemma 1, we obtain

$$\begin{bmatrix}\Delta A, & \beta(\Delta B)^T, & \lambda\Delta f\end{bmatrix}^T = \begin{bmatrix}\widetilde{u}\\ \frac{1}{\beta}\widetilde{p}\\ -\frac{1}{\lambda}\end{bmatrix}^{\dagger^T} r_f^T + \left(I - \begin{bmatrix}\widetilde{u}\\ \frac{1}{\beta}\widetilde{p}\\ -\frac{1}{\lambda}\end{bmatrix}\begin{bmatrix}\widetilde{u}\\ \frac{1}{\beta}\widetilde{p}\\ -\frac{1}{\lambda}\end{bmatrix}^{\dagger}\right) W,$$

$$\begin{bmatrix}\alpha\Delta B, & \mu\Delta g\end{bmatrix}^T = \begin{bmatrix}\frac{1}{\alpha}\widetilde{u}\\ -\frac{1}{\mu}\end{bmatrix}^{\dagger^T} r_g^T + \left(I - \begin{bmatrix}\frac{1}{\alpha}\widetilde{u}\\ -\frac{1}{\mu}\end{bmatrix}\begin{bmatrix}\frac{1}{\alpha}\widetilde{u}\\ -\frac{1}{\mu}\end{bmatrix}^{\dagger}\right) Z,$$

where $W \in \mathbb{R}^{(m+n+1)\times m}$, $Z \in \mathbb{R}^{(m+1)\times n}$. Therefore,

$$\begin{aligned}
&\left[\eta^{(\theta,\ \lambda,\ \mu)}(\widetilde{u},\widetilde{p})\right]^2\\
&= \min_{\substack{Z,W\\ \alpha^2+\beta^2=\theta^2}} \left\{ \left\|[\Delta A, \beta(\Delta B)^T, \lambda\Delta f]\right\|_F^2 + \left\|[\alpha(\Delta B)^T, \mu\Delta g]\right\|_F^2 \right\}\\
&\geqslant \min_{\alpha^2+\beta^2=\theta^2} \left\{ \min_W \left\|[\Delta A, \beta(\Delta B)^T, \lambda\Delta f]\right\|_F^2 + \min_Z \left\|[\alpha(\Delta B)^T, \mu\Delta g]\right\|_F^2 \right\}\\
&= \min_{\alpha^2+\beta^2=\theta^2} \left\{ \frac{\|r_f\|^2}{\beta^{-2}\|\widetilde{p}\|^2 + \|\widetilde{u}\|^2 + \lambda^{-2}} + \frac{\|r_g\|^2}{\alpha^{-2}\|\widetilde{u}\|^2 + \mu^{-2}} \right\}\\
&= \frac{(\|\widetilde{u}\|\|r_f\| - \|\widetilde{p}\|\|r_g\|)^2 + \theta^2\mu^{-2}\|r_f\|^2 + \theta^2(\|\widetilde{u}\|^2 + \lambda^{-2})\|r_g\|^2}{(\|\widetilde{u}\|^2 + \theta^2\mu^{-2})(\|\widetilde{u}\|^2 + \lambda^{-2}) + \mu^{-2}\|\widetilde{p}\|^2}.
\end{aligned}$$

Supposing $\theta > 1$, we have

$$\begin{aligned}
\frac{\eta_S(\widetilde{u},\widetilde{p})}{\eta(\widetilde{z})} &\geqslant \left( \frac{\theta^2(\|\widetilde{u}\|^2 + \lambda^{-2})\|r_g\|^2}{(\|\widetilde{u}\|^2 + \theta^2\mu^{-2})(\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2 + \lambda^{-2})} \frac{\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2}{\|r_f\|^2 + \|r_g\|^2} \right)^{\frac{1}{2}}\\
&= \theta\sqrt{\frac{\|\widetilde{u}\|^2 + \lambda^{-2}}{\|\widetilde{u}\|^2 + \theta^2\mu^{-2}}}\sqrt{\frac{\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2}{\|\widetilde{u}\|^2 + \|\widetilde{p}\|^2 + \lambda^{-2}}}\sqrt{\frac{\|r_g\|^2}{\|r_f\|^2 + \|r_g\|^2}}.
\end{aligned}$$

If $\theta \gg 1$, and

$$\|r_f\| \sim \|r_g\|, \theta\mu^{-1} \sim \mathcal{O}(1), \lambda \sim \mathcal{O}(1), \tag{3.1}$$

then

$$\frac{\eta_S(\widetilde{u},\widetilde{p})}{\eta(\widetilde{z})} \sim \mathcal{O}(\theta) \gg 1,$$

which shows that the structured backward error can be arbitrarily larger than the unstructured one.

## 4. Numerical examples.

In this section we will examine two numerical cases and compare the normwise structured and unstructured backward errors.

[illegible] 1. Consider the saddle point systems with

$$B = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0\\ 10^{-3} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad A = DMD,$$

$$M = \text{magic}(6) + \text{eye}(6), \qquad D = \text{diag}([1, 5, 10, 50, 100, 10000]),$$

$$f = [10^8, 10, 0, 0, 0, 0]^T, \qquad g = [10^{-8}, 0, 0]^T.$$

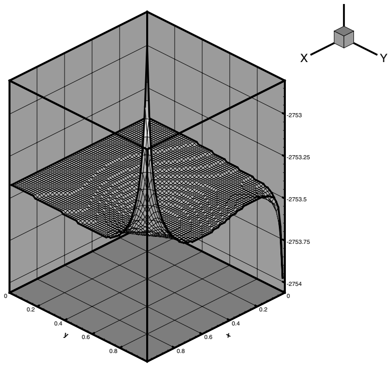


FIG. 1. *Three-dimensional surface plot of pressure.*

This unsymmetric case is slightly adapted from the example in [20]. We use the MATLAB backslash function (or matrix left division) to solve this problem and obtain the following computed solutions:

$$\widetilde{u} = \begin{pmatrix} -3.430234778891480 \times 10^{-09} \\ 4.517200836796800 \times 10^{-10} \\ 9.834679521009126 \times 10^{-09} \\ -1.139022982251766 \times 10^{-09} \\ -3.743828054634186 \times 10^{-10} \\ -1.254493123951351 \times 10^{-11} \end{pmatrix},$$

$$\widetilde{p} = \begin{pmatrix} 4.563714591880652 \times 10^{-05} \\ 1.000002220308359 \times 10^{+01} \\ 1.000000000000047 \times 10^{+11} \end{pmatrix}.$$

By (1.3) we get the unstructured backward error

$$\eta(\widetilde{z}) = 4.0 \times 10^{-30}.$$

But the structured backward error is given by

$$\eta_S(\widetilde{u}, \widetilde{p}) = 2.7 \times 10^{-2}.$$

We can see that Gaussian elimination with partial pivoting for solving this saddle point systems is backward stable but not strongly stable.

[illegible] 2. Consider the impressible flow in $\Omega = (0,1) \times (0,1)$, with Dirichlet boundary conditions on $\partial\Omega$. Let $\mathbf{u} = (u_1, u_2)^T$ denote the velocity field and $p$ the pressure. On the boundaries the velocities are zeros except for the horizontal velocity, which is $u_1 = 1$ on the upper boundary. The governing equations are Navier–Stokes equations:

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p &= 0, \\ -\nabla \cdot \mathbf{u} &= 0. \end{aligned}$$

Using the MAC finite difference scheme, where the discrete velocities and pressures are defined on staggered grids, we have the saddle point systems (1.1) with $g = 0$.

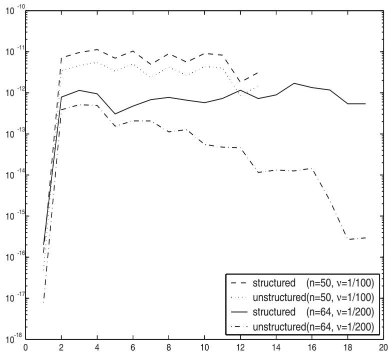


FIG. 2. *Backward error.*

In practical computation, we take $64 \times 64$ grids, choose $\nu = 1/200$, and apply GMRES [17] to solve (1.1). After 19 Picard iterations, we obtain the converged pressure field as shown in Figure 1. Using (2.8) the structured backward error of the last Picard iteration is $5.43 \times 10^{-13}$, while the unstructured backward error is $2.96 \times 10^{-16}$ (see Figure 2). Though the structured backward error is about three orders larger than the unstructured one, they are both small. When $\nu$ becomes larger, the difference between structured and unstructured backward error is smaller. For example, if we take $\nu = 1/100$ and $50 \times 50$ grids, the structured backward error is only two times the unstructured one at each Picard iteration, as shown in Figure 2. We can conclude that the algorithm is strongly stable for this problem.

**Acknowledgments.** The authors would like to thank the editor Andy Wathen and two referees for their very useful and detailed suggestions on our early version.

## REFERENCES

[1] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
[2] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
[3] J. R. Bunch, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Algebra Appl., 88/89 (1987), pp. 49–66.
[4] J. R. Bunch, J. W. Demmel, and C. F. Van Loan, *The strong stability of algorithms for solving symmetric linear systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 494–499.
[5] L. Eldén, *Perturbation theory for the least squares problem with linear equality constraints*, SIAM J. Numer. Anal., 17 (1980), pp. 338–350.
[6] H. Elman and D. Silvester, *Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations*, SIAM J. Sci. Comput., 17 (1996), pp. 33–46.
[7] H. C. Elman, *Preconditioning for the steady-state Navier–Stokes equations with low viscosity*, SIAM J. Sci. Comput., 20 (1999), pp. 1299–1316.
[8] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, UK, 2005.
[9] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[10] M. Gulliksson, X. Jin, and Y. Wei, *Perturbation bounds for constrained and weighted least squares problems*, Linear Algebra Appl., 349 (2002), pp. 221–232.
[11] M. Gulliksson and P. Å. Wedin, *Perturbation theory for generalized and constrained linear least squares*, Numer. Linear Algebra Appl., 7 (2000), pp. 181–195.
[12] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[13] X. Li and X. Liu, *Structured backward errors for structured KKT systems*, J. Comput. Math., 22 (2004), pp. 605–610.

[14] Y. Lin and Y. Wei, *Fast corrected Uzawa methods for solving symmetric saddle point problems*, Calcolo, 43 (2006), pp. 65–82.

[15] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numer. Math., 6 (1964), pp. 405–409.

[16] J. L. Rigal and J. Gaches, *On the compatibility of a given solution with data of a linear system*, J. Assoc. Comput. Mach., 14 (1967), pp. 543–548.

[17] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[18] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA, 1986.

[19] G. Strang, *A framework for equilibrium equations*, SIAM Rev., 30 (1988), pp. 283–297.

[20] J.-G. Sun, *Structured backward errors for KKT systems*, Linear Algebra Appl., 288 (1999), pp. 75–88.

[21] J.-G. Sun, *A note on backward errors for structured linear systems*, Numer. Linear Algebra Appl., 12 (2005), pp. 585–603.

[22] S. A. Vavasis, *Stable numerical algorithms for equilibrium systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1108–1131.

[23] G. Wang, Y. Wei, and S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, Beijing, 2004.

[24] M. Wei, *Perturbation theory for the rank-deficient equality constrained least squares problem*, SIAM J. Numer. Anal., 29 (1992), pp. 1462–1481.

[25] M. H. Wright, *Interior methods for constrained optimization*, in Acta Numerica 1992, Acta Numer., Cambridge University Press, Cambridge, UK, 1992, pp. 341–407.

[26] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

[27] H. Xiang, *Iterative Methods and Perturbation Analysis of Structured Linear Systems*, Ph.D. thesis, School of Mathematical Sciences, Fudan University, Shanghai, China, 2006.

[28] H. Xiang, Y. Wei, and H. Diao, *Perturbation analysis of generalized saddle point systems*, Linear Algebra Appl., 419 (2006), pp. 8–23.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 3, pp. 850–854

# UNITARILY INVARIANT NORMS OF TOEPLITZ MATRICES WITH FISHER–HARTWIG SINGULARITIES*

SEAK-WENG VONG† AND XIAO-QING JIN†

**Abstract.** We provide upper bounds for unitarily invariant norms of finite Toeplitz matrices generated by functions with a Fisher–Hartwig singularity. These bounds are sharp as the matrix size goes to infinity. Our results improve previous estimates and confirm a conjecture recently raised by Böttcher.

**Key words.** unitarily invariant norm, Schatten norm, Ky Fan norm, Toeplitz matrix, Fisher–Hartwig singularity

**AMS subject classifications.** 47B35, 15A60, 65F10

**DOI.** 10.1137/070681065

## 1. Introduction.

Denote by $T_n(a)$ the $n \times n$ Toeplitz matrix $(a_{j-k})_{j,k=1}^n$, where

$$a_m = \frac{1}{2\pi}\int_{-\pi}^{\pi} a(x)e^{-imx}dx, \qquad m = 0, \pm 1, \pm 2, \ldots,$$

are the Fourier coefficients of a function $a \in L^1(-\pi,\pi)$. In this paper, we consider asymptotic properties of unitarily invariant norms of $T_n(a)$ when $a$ has a singularity of the Fisher–Hartwig type. An archetypal example [2, 3] of such a function is

$$\omega_\alpha^+(x) = \begin{cases} 0, & x \in (-\pi, 0), \\ x^{-\alpha}, & x \in (0, \pi), \end{cases} \tag{1}$$

where $0 < \alpha < 1$.

Now, we introduce a class of unitarily invariant norms called the Schatten $p$-norms [1]. For $1 \le p < \infty$, the Schatten $p$-norms $\|T_n(a)\|_p$ are defined by

$$\|T_n(a)\|_p \equiv \left[\sum_{j=1}^n s_j^p(T_n(a))\right]^{1/p}, \tag{2}$$

and $\|T_n(a)\|_\infty$ is defined as $s_1(T_n(a))$, where $s_1(T_n(a)) \ge \cdots \ge s_n(T_n(a))$ are the singular values of $T_n(a)$. For $\omega_\alpha^+$ given by (1), one can compute the Frobenius norm, which is just the case $p = 2$ in (2), to obtain that

$$\|T_n(\omega_\alpha^+)\|_2 \le \begin{cases} C_2(\alpha)n^{1/2}, & \alpha < 1/2, \\ C_2(\alpha)(n\log n)^{1/2}, & \alpha = 1/2, \\ C_2(\alpha)n^\alpha, & \alpha > 1/2, \end{cases} \tag{3}$$

---

*Received by the editors January 26 , 2007; accepted for publication (in revised form) by M. Benzi February 5, 2007; published electronically July 25, 2007.
http://www.siam.org/journals/simax/29-3/68106.html

†Department of Mathematics, University of Macau, Macao, China (swvong@umac.mo, xqjin@umac.mo).

where $C_2(\alpha)$ is a positive constant depending on $\alpha$ [2].

Note that if we put $\omega_\alpha^-(x) = \omega_\alpha^+(-x)$, then functions of the form

$$a(x) = \omega_\alpha^-(x)b(x) + \omega_\alpha^+(x)c(x),$$

where $b$, $c \in L^\infty$, include all functions with a Fisher–Hartwig singularity. It is proved in [2] that for a given $0 < \alpha < 1$, one has the following bound for the Schatten $p$-norms:

$$\|T_n(a)\|_p \leq \begin{cases} C_p(a)n^\alpha(\log n)^{1+\alpha}, & p = 1/\alpha, \\ C_p(a)n^\alpha \log n, & 1/\alpha < p < \infty, \\ C_p(a)n^\alpha, & p = \infty, \end{cases} \tag{4}$$

where $C_p(a)$ is a positive constant depending on $p$ and $a$.

Comparing the above two estimates (3) and (4), it seems that there is room for improvement. Böttcher [2] conjectured that the $n^\alpha(\log n)^{1+\alpha}$ in (4) can be replaced by $(n \log n)^\alpha$ and that the $n^\alpha \log n$ in (4) can be improved to $n^\alpha$. The proof of these conjectures is the objective of this paper. We would like to mention that the results in [2] were obtained through a subtle analysis on the Dirichlet kernel. Our method goes in another direction and depends mainly on a basic theory of unitarily invariant norms.

It is well known [1] that a function $\|\cdot\|$ defined on $n \times n$ matrices is a unitarily invariant norm if and only if there exists a symmetric gauge function $\Phi$ on $\mathbf{R}^n$ such that

$$\|A\| = \Phi(s_1(A), \ldots, s_n(A)).$$

We recall that a function $\Phi : \mathbf{R}^n \to \mathbf{R}_+$ is called a symmetric gauge function if it satisfies the following properties:

(i) $\Phi$ is a norm.
(ii) $\Phi(Px) = \Phi(x)$ for all $x \in \mathbf{R}^n$ and all permutation matrices $P$.
(iii) $\Phi(\varepsilon_1 x_1, \ldots, \varepsilon_n x_n) = \Phi(x_1, \ldots, x_n)$ if $\varepsilon_j = \pm 1$ for any $j$.
(iv) $\Phi(1, 0, \ldots, 0) = 1$.

In the following discussion, we use $\|\cdot\|_\Phi$ to denote the unitarily invariant norm corresponding to a given symmetric gauge function $\Phi$.

The paper is organized as follows. In section 2, by using the Fan dominance theorem and Ky Fan $k$-norms, a general estimate on unitarily invariant norms is established. In section 3, we apply the estimate obtained in section 2 to get an improved bound on the Schatten $p$-norms.

## 2. An estimate on unitarily invariant norms.

We first concentrate on $\omega_\alpha^\pm(x)$. Our analysis depends mainly on the following theorem [1].

Theorem 1 (Fan dominance theorem). [illegible] $A, B$ [illegible] $n \times n$ [illegible]

$$\|A\|_{(k)} \leq \|B\|_{(k)} \qquad [illegible] \quad k = 1, 2, \ldots, n,$$

[illegible]

$$|||A||| \leq |||B|||$$

[illegible]

$$\|A\|_{(k)} \equiv \sum_{j=1}^{k} s_j(A)$$

[illegible] $k$ [illegible]

Theorem 1 implies that in order to compare any given unitarily invariant norm of two matrices, one needs only to compare the Ky Fan $k$-norms. This gives the importance of the Ky Fan $k$-norms among unitarily invariant norms and inspires one to estimate the Ky Fan $k$-norms of $\omega_\alpha^\pm$. Using that $T_n(h)$ may be interpreted as a compression of the operator of multiplication by $h$ on $L^2(-\pi,\pi)$, we see that $s_1(T_n(h)) = \|T_n(h)\|_\infty \le \|h\|_\infty = h_{\max}$. This implies the following.

THEOREM 2. [illegible] $1 \le k \le n$ [illegible]

$$\|T_n(\omega_\alpha^\pm)\|_{(k)} \le \frac{(2\pi)^{-1} + 1 - \alpha}{1-\alpha}\, k^{1-\alpha} n^\alpha \le \frac{C}{1-\alpha}\, k^{1-\alpha} n^\alpha.$$

[illegible] $C$ [illegible] $n$, $k$ [illegible] $\alpha$

[illegible]. We need only to work out the proof for $\omega_\alpha^+(x)$. The case for $\omega_\alpha^-(x)$ is similar. For a fixed $k$, we write

$$\omega_\alpha^+(x) = g(x) + h(x),$$

where

$$g(x) = \begin{cases} 0, & x \in (-\pi, 0), \\ x^{-\alpha}, & x \in (0, k/n), \\ 0, & x \in (k/n, \pi), \end{cases} \qquad (g \in L^1),$$

$$h(x) = \begin{cases} 0, & x \in (-\pi, k/n), \\ x^{-\alpha}, & x \in (k/n, \pi), \end{cases} \qquad (h \in L^\infty).$$

We clearly have

$$T_n(\omega_\alpha^+) = T_n(g) + T_n(h).$$

This implies that

$$\begin{aligned} &\|T_n(\omega_\alpha^+)\|_{(k)} \le \|T_n(g)\|_{(k)} + \|T_n(h)\|_{(k)} \\ &\le \sum_{j=1}^{n} s_j(T_n(g)) + \sum_{j=1}^{k} s_j(T_n(h)) \\ &\le \operatorname{tr}(T_n(g)) + k s_1(T_n(h)) \le \frac{n}{2\pi}\int_{-\pi}^{\pi} g(x)dx + k h_{\max} \\ &= \frac{(2\pi)^{-1} n}{1-\alpha}\left(\frac{k}{n}\right)^{1-\alpha} + k\left(\frac{k}{n}\right)^{-\alpha} = \frac{(2\pi)^{-1} + 1 - \alpha}{1-\alpha} k^{1-\alpha} n^\alpha. \qquad \square \end{aligned}$$

As mentioned above, a function with a Fisher–Hartwig singularity of degree $\alpha$ can be written as

$$a(x) = b(x)\omega_\alpha^-(x) + c(x)\omega_\alpha^+(x).$$

By the fact that

$$|a(x)| \leq \|b\|_\infty \omega_\alpha^-(x) + \|c\|_\infty \omega_\alpha^+(x),$$

we thus have [4]

$$\|T_n(a)\|_{(k)} \leq \|b\|_\infty \|T_n(\omega_\alpha^-)\|_{(k)} + \|c\|_\infty \|T_n(\omega_\alpha^+)\|_{(k)}.$$

This gives by Theorem 2 the following.

THEOREM 3. [illegible] $a(x)$ [illegible] $\alpha$ [illegible]

$$\|T_n(a)\|_{(k)} \leq \frac{C}{1-\alpha} k^{1-\alpha} n^\alpha, \qquad 1 \leq k \leq n.$$

A direct consequence of Theorems 1 and 2 is an estimate on unitarily invariant norms of $T_n(\omega_\alpha^\pm)$. To this end, we compare the Ky Fan $k$-norms of $T_n(\omega_\alpha^\pm)$ and that of the diagonal matrix

$$D_\alpha \equiv \frac{(2\pi)^{-1} + 1 - \alpha}{1-\alpha} n^\alpha \operatorname{diag}(1, 2^{1-\alpha}-1, \ldots, k^{1-\alpha}-(k-1)^{1-\alpha}, \ldots, n^{1-\alpha}-(n-1)^{1-\alpha}).$$

By Theorem 2, it is easy to see that

$$\|T_n(\omega_\alpha^\pm)\|_{(k)} \leq \|D_\alpha\|_{(k)}$$

for $k = 1, 2, \ldots, n$. Recall that every unitarily invariant norm is induced by a symmetric gauge function $\Phi$ on $R^n$. We thus conclude the following.

THEOREM 4. [illegible] $\|\cdot\|_\Phi$ [illegible] $\Phi$ [illegible] $\mathbf{R}^n$ [illegible]

$$\|T_n(\omega_\alpha^\pm)\|_\Phi \leq \|D_\alpha\|_\Phi = \frac{(2\pi)^{-1} + 1 - \alpha}{1-\alpha} n^\alpha$$

$$\times \Phi(1, 2^{1-\alpha} - 1, \ldots, k^{1-\alpha} - (k-1)^{1-\alpha}, \ldots, n^{1-\alpha} - (n-1)^{1-\alpha}).$$

**3. Application to the Schatten *p*-norms.** Recall the definition of the Schatten $p$-norms defined by (2). By Theorem 4, in order to estimate $\|T_n(\omega_\alpha^\pm)\|_p$, we need to get an upper bound for

$$\left[\sum_{k=1}^n (k^{1-\alpha} - (k-1)^{1-\alpha})^p\right]^{1/p}.$$

Using that $\varphi(k) - \varphi(k-1) = \varphi'(\xi_k)$ with $k-1 \leq \xi_k \leq k$, we see that, for $2 \leq k \leq n$,

$$0 < k^{1-\alpha} - (k-1)^{1-\alpha} = (1-\alpha)\xi_k^{-\alpha} \leq (1-\alpha)(k-1)^{-\alpha}.$$

Hence, Theorem 4 gives

$$\|T_n(\omega_\alpha^+)\|_p \leq \frac{Cn^\alpha}{1-\alpha}\left(1 + \sum_{k=1}^{n-1} k^{-\alpha p}\right)^{1/p} \leq \frac{Cn^\alpha}{1-\alpha}\left(2 + \int_1^{n-1} x^{-\alpha p} dx\right)^{1/p}.$$

The right-hand side can be bounded with respect to $p$ as

$$\begin{cases} C\dfrac{1}{(1-\alpha)(1-\alpha p)^{1/p}}n^{1/p}, & \alpha p < 1, \\[2ex] C\dfrac{1}{1-\alpha}\,n^{\alpha}(\log n)^{1/p}, & \alpha p = 1, \\[2ex] C\dfrac{1}{(1-\alpha)(\alpha p-1)^{1/p}}n^{\alpha}, & \alpha p > 1. \end{cases}$$

Similar to the discussion given in the case of the Ky Fan $k$-norms, we get the following.

THEOREM 5. [illegible] $a(x)$ [illegible] $\alpha$ [illegible] $1 \le p < \infty$ [illegible]

$$\|T_n(a)\|_p \le \begin{cases} C\dfrac{1}{(1-\alpha)(1-\alpha p)^{1/p}}\,n^{1/p}, & \alpha p < 1, \\[2ex] C\dfrac{1}{1-\alpha}\,(n\log n)^{\alpha}, & \alpha p = 1, \\[2ex] C\dfrac{1}{(1-\alpha)(\alpha p-1)^{1/p}}\,n^{\alpha}, & \alpha p > 1. \end{cases}$$

[illegible] Theorem 5 is clearly better than (4) and is consistent with (3). This theorem confirms a conjecture raised in [2]. Moreover, the method we used here is elementary.

## REFERENCES

[1] R. BHATIA, *Matrix Analysis*, Springer-Verlag, New York, 1997.

[2] A. BÖTTCHER, *Schatten norms of Toeplitz matrices with Fisher-Hartwig singularities*, Electron. J. Linear Algebra, 15 (2006), pp. 251–259.

[3] A. BÖTTCHER AND J. VIRTANEN, *Norms of Toeplitz matrices with Fisher–Hartwig symbols*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 660–671.

[4] S. SERRA AND P. TILLI, *On unitarily invariant norms of matrix-valued linear positive operators*, J. Inequal. Appl., 7 (2002), pp. 309–330.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 855–869

# NUMERICALLY STABLE IMPLEMENTATIONS OF THE STRUCTURED COVARIANCE EXPECTATION-MAXIMIZATION ALGORITHM*

DANIEL R. FUHRMANN†

**Abstract.** Numerically stable methods for the computations required in the expectation-maximization (EM) algorithm for maximum-likelihood structured covariance estimation are presented. It is shown that the basic computational task at each iteration is the calculation of a pseudoinverse of a certain linear system of equations. In the no-noise case this is a hard-decision, or Moore–Penrose, pseudoinverse, whereas in the additive noise case it is a soft-decision pseudoinverse. An approach to computing the soft-decision pseudoinverse, which can handle all combinations of dimension and rank in this system of equations, based on the singular value decomposition (SVD), is proposed. An alternative method based on the LQ factorization, which is applicable in certain circumstances, is also proposed. The intermediate calculations required in the EM algorithm can be used to calculate the log-likelihood and the gradient of the log-likelihood.

**Key words.** structured covariance estimation, spectrum estimation, signal processing, EM algorithm, SVD

**AMS subject classifications.** 62H12, 65F30

**DOI.** 10.1137/040609495

## 1. Introduction.

The estimation of the second-order statistics of time series or sensor array data is a central problem in statistical signal processing. When the model for the data involves a linear transformation of a large number of independent components, or perhaps of a continuous independent-increments process, then the problem is one of spectrum estimation. When one considers the variance or covariance of the discrete data themselves, as determined by the spectrum, then the problem is one of structured covariance estimation. These problems are obviously closely related and have been well studied; see, e.g., [1, 2, 3, 4, 5, 6] and the many references contained therein.

The canonical problem in this class is one in which the spectrum denotes the power or power density of independent sinusoidal or complex exponential components, and the resulting time series has a Toeplitz covariance structure [6, 7, 8]. However, there are many generalizations and extensions of this basic problem. In problems involving space, or space and time, the spectrum could represent power or power density as a function of spatial variables or angle of arrival [9, 10]. The sensors could be placed in some arbitrary geometry and have relatively unstructured response vectors [11]. The relationship between the underlying independent-increments process and the data may be nonstationary, as in radio astronomy or time-varying arrays [12, 13]. In the radar imaging problem described in [14, 15], the scattering function parameters are indexed

---

[*] Received by the editors June 4, 2004; accepted for publication (in revised form) by H. J. Werner March 5, 2007; published electronically August 1, 2007. This work was supported by the U.S. Defense Advanced Research Projects Agency (DARPA) under agreement F30602-03-2-0043. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

http://www.siam.org/journals/simax/29-3/60949.html

[†] Electronics Systems and Signals Research Laboratory, Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130 (danf@ese.wustl.edu).

by the radar-centered parameters of range and Doppler, leading to the estimation of a block-diagonal matrix with independent Toeplitz blocks. In our recent work in radar imaging [16], we showed how the scattering function may be estimated in an active way, with the illumination under the control of the data collection instrumentation.

When the underlying spectrum is discretized, many of these problems are based on the complex multivariate statistical model[1]

$$\mathbf{y}_k \sim CN(0, \mathbf{R}_k), \tag{1.1}$$

where

$$\mathbf{R}_k = \mathbf{A}_k \Sigma \mathbf{A}_k^{\mathrm{H}}. \tag{1.2}$$

The maximum-likelihood estimation of the diagonal matrix $\Sigma$ given the data $\mathbf{y}_1 \ldots \mathbf{y}_K$ is an interesting statistical inference problem with no known closed-form solution. The expectation-maximization (EM) algorithm [17] has been developed and implemented for a number of versions of this problem [7, 9, 10, 11, 12, 13, 14, 15, 16].

Our concern here is with the computational aspects of the steps of the EM algorithm. Recent work in scattering function estimation shows that one can have, in the same problem, several different matrices $\mathbf{A}_k$ which can be either wide or long[2] and can be either full-rank or rank-deficient. The standard derivations of the structured covariance EM algorithm assume that $\mathbf{A}_k$ is wide and full-rank. Other possibilities do not invalidate the basic maximum-likelihood approach; they simply require more care. This paper presents a computational approach that handles all combinations of dimension and rank in a numerically stable[3] way. It is based on the singular value decomposition (SVD) of the matrix $\mathbf{A}_k \Sigma^{\frac{1}{2}}$, which can be viewed as a square root of the covariance $\mathbf{R}_k$. Although the SVD itself is often seen as computationally expensive, calculations that are based on it are straightforward and have an intuitive appeal. Computational methods for the SVD are well understood, have been optimized over several decades of numerical linear algebra research and software development, and may not be the bottleneck commonly presumed.

In the standard case ($\mathbf{A}$ wide and full-rank), a computationally efficient method can be built on the LQ factorization rather than the SVD. Both the SVD and the LQ approaches have the advantage of not requiring data-squaring or sample covariance calculations; algorithms such as these are often called data-domain algorithms.

The quantities computed in the course of the structured covariance EM algorithm are essentially the same as those needed for the computation of the log-likelihood and the gradient of the log-likelihood with respect to the spectrum. Thus the algorithms presented here may have applicability beyond the EM algorithm itself.

Section 2 reviews the maximum-likelihood estimation problem and the derivation of the EM algorithm, for both the no-noise and additive noise cases. Section 3 shows how the basic computational task can be interpreted as either a hard-decision (Moore–Penrose) or a soft-decision pseudoinverse, for the no-noise and additive noise cases,

---

[1] (1.1) is shorthand for "$\mathbf{y}_k$ is subject to a complex multivariate Gaussian distribution with mean 0 and covariance $\mathbf{R}_k$," and this notation will be used throughout. The superscript H denotes Hermitian transpose.

[2] An $M \times N$ matrix $\mathbf{A}$ is said to be *wide* if $M < N$, *square* if $M = N$, and *long* if $M > N$.

[3] By "numerically stable" it is meant simply that the numerical or round-off behavior does not deteriorate badly as the rank of $\mathbf{A}$ goes to 0 or as $\mathbf{R}$ becomes singular. This paper contains no formal round-off analysis of the proposed methods; rather, we appeal to the favorable numerical properties of SVD and LQ factorizations and the least-squares methods based on them.

respectively. Section 4 shows how the pseudoinverses can be computed with the aid of the SVD and LQ factorizations. Finally, we show how the intermediate computations can be applied to the calculation of the log-likelihood and the gradient of the log-likelihood.

## 2. Structured covariance estimation and the EM algorithm.

**2.1. Basic model.** We begin with a brief review of the structured covariance estimation problem and an abbreviated derivation of the EM algorithm.

Suppose that there exist $K$ independent and identically distributed (i.i.d.) complex Gaussian random vectors $\mathbf{x}_1 \dots \mathbf{x}_K$, with mean 0 and diagonal covariance $\Sigma$. This is denoted by

$$\mathbf{x}_k \sim CN(0, \Sigma), \quad k = 1 \dots K, \text{ i.i.d.,} \tag{2.1}$$

where

$$\Sigma = \operatorname{diag}(\sigma(1) \dots \sigma(N)). \tag{2.2}$$

The diagonal elements of $\Sigma$ can be thought of as a discrete power spectrum that we wish to estimate.[4] If the $\mathbf{x}_k$ were observable, the sufficient statistic for $\sigma(n)$ would be

$$\tau(n) = \frac{1}{K} \sum_{k=1}^{K} |x_k(n)|^2, \quad n = 1 \dots N, \tag{2.3}$$

and the maximum-likelihood estimate of $\sigma(n)$ would be equal to $\tau(n)$.

Suppose now that, instead of observing the $\mathbf{x}_k$ directly, one is only able to observe the $\mathbf{x}_k$ through a linear transformation $\mathbf{A}$. That is, the data are complex Gaussian random vectors $\mathbf{y}_k$, $k = 1 \dots K$, described by

$$\mathbf{y}_k = \mathbf{A}\mathbf{x}_k. \tag{2.4}$$

It follows that the distribution of the $\mathbf{y}_k$ is

$$\mathbf{y}_k \sim CN(0, \mathbf{R}), \tag{2.5}$$

where

$$\mathbf{R} = \mathbf{A}\Sigma\mathbf{A}^{\mathrm{H}}. \tag{2.6}$$

The maximum-likelihood estimation problem is to estimate $\Sigma$ from the $\mathbf{y}_k$, which is generally a much more difficult problem than estimating $\Sigma$ from the $\mathbf{x}_k$. We refer to the $\mathbf{x}_k$ as the [illegible] and the $\mathbf{y}_k$ as the [illegible]. Because $\mathbf{R}$, the covariance for $\mathbf{y}_k$, is constrained to belong to a proper subset of the space of the nonnegative Hermitian matrices, as given in (2.6), the estimation problem is often called [illegible].

The sufficient statistic for $\mathbf{R}$, as in any covariance estimation problem, is the sample covariance given by

$$\mathbf{S} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{y}_k \mathbf{y}_k^{\mathrm{H}}. \tag{2.7}$$

---

[4] Although $\sigma$ is commonly used for standard deviation, here we use it for variance to maintain a consistency in notation for $\sigma$ and $\Sigma$.

The incomplete-data log-likelihood, with constant terms removed, is

$$l(\mathbf{R};\mathbf{S}) = -K\log\det\mathbf{R} - K\mathrm{tr}\mathbf{R}^{-1}\mathbf{S}. \tag{2.8}$$

The maximum-likelihood estimation problem is to maximize (2.8) with respect to either $\Sigma$ or the constrained $\mathbf{R}$. It is usually convenient to cast the optimization problem as one of searching in $\Sigma$-space.

The following two identities, stated here without proof (see [6]), are useful in determining the gradient of the log-likelihood and thus the necessary conditions for a maximizer. They are

$$\delta\log\det\mathbf{R} = \mathrm{tr}\mathbf{R}^{-1}\delta\mathbf{R} \tag{2.9}$$

and

$$\delta\mathbf{R}^{-1} = -\mathbf{R}^{-1}\delta\mathbf{R}\mathbf{R}^{-1}, \tag{2.10}$$

where the symbol $\delta$ denotes the first variation. From (2.6) it is clear that

$$\delta\mathbf{R} = \mathbf{A}\delta\Sigma\mathbf{A}^{\mathrm{H}}. \tag{2.11}$$

It follows then that the variation of the log-likelihood is given by

$$\delta l = \mathrm{tr}\mathbf{A}^{\mathrm{H}}(\mathbf{R}^{-1}\mathbf{S}\mathbf{R}^{-1} - \mathbf{R}^{-1})\mathbf{A}\delta\Sigma. \tag{2.12}$$

One way to interpret (2.12) is that the diagonal elements of $\mathbf{A}^{\mathrm{H}}(\mathbf{R}^{-1}\mathbf{S}\mathbf{R}^{-1} - \mathbf{R}^{-1})\mathbf{A}$ form a gradient vector $\mathbf{g}$ for the parameter vector $\sigma$. The $\sigma(n)$ are nonnegative but are otherwise unconstrained. It follows that the Kuhn–Tucker conditions for a maximizer of the log-likelihood are (a) for any $\sigma(n) \neq 0$, the corresponding $g_n = 0$, and (b) for any $\sigma(n) = 0$, the corresponding $g_n \leq 0$.

The EM algorithm [17] for the structured covariance estimation problem is an iterative algorithm whose stationary points satisfy the Kuhn–Tucker conditions just given. Define $\Sigma^{(p)}$ as the estimate of $\Sigma$ at iteration $p$. Each iteration of the EM algorithm comprises two steps, the E-step (expectation) and the M-step (maximization). The E-step is to compute the conditional expected value of the complete-data sufficient statistics, given the current estimate $\Sigma^{(p)}$ and the incomplete-data sufficient statistic $\mathbf{S}$. In this case, the complete-data sufficient statistics are the $s(n)$ as given in (2.3). The M-step is the trivial operation of setting the next iterate $\sigma^{(p)}(n)$ equal to the estimated $s(n)$.

The expected squared magnitude of $x_k(n)$ is the squared magnitude of the conditional mean, plus the conditional variance. The conditional mean is given by

$$\begin{aligned}\mathrm{E}\{\mathbf{x}_k|\mathbf{y}_k,\Sigma^{(p)}\} &= \mathbf{R}_{\mathbf{xy}}(p)\mathbf{R}_{\mathbf{yy}}^{-1}(p)\mathbf{y}_k \\ &= \Sigma^{(p)}\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathbf{yy}}^{-1}(p)\mathbf{y}_k,\end{aligned} \tag{2.13}$$

where $\mathbf{R}_{\mathbf{yy}}(p)$ is the covariance of $\mathbf{y}_k$ at iteration $p$, and $\mathbf{R}_{\mathbf{xy}}(p)$ is the cross-covariance of $\mathbf{x}_k$ and $\mathbf{y}_k$, both constructed with the assumed value of $\Sigma^{(p)}$.

The conditional covariance of the $\mathbf{x}_k$ is given by

$$\begin{aligned}cov\{\mathbf{x}_k|\mathbf{y}_k,\Sigma^{(p)}\} &= \mathbf{R}_{\mathbf{xx}}(p) - \mathbf{R}_{\mathbf{xy}}(p)\mathbf{R}_{\mathbf{yy}}^{-1}(p)\mathbf{R}_{\mathbf{yx}} \\ &= \Sigma - \Sigma^{(p)}\mathbf{A}^{\mathrm{H}}\mathbf{R}_{\mathbf{yy}}^{-1}(p)\mathbf{A}\Sigma^{(p)}.\end{aligned} \tag{2.14}$$

Substituting these conditional expectations in place of the actual $\mathbf{x}_k$ in (2.3), and using the fact that

$$\mathbf{R}_{\mathbf{yy}}(p) = \mathbf{R}(p) = \mathbf{A}\Sigma^{(p)}\mathbf{A}^{\mathrm{H}}, \tag{2.15}$$

we have finally the EM iteration given by

$$\begin{aligned}\Sigma^{(p+1)} = \Sigma^{(p)} &+ \operatorname{diag}\left[\Sigma^{(p)}\mathbf{A}^{\mathrm{H}}\mathbf{R}^{-1}(p)\mathbf{S}\mathbf{R}^{-1}(p)\mathbf{A}\Sigma^{(p)}\right] \\ &- \operatorname{diag}\left[\Sigma^{(p)}\mathbf{A}^{\mathrm{H}}\mathbf{R}^{-1}(p)\mathbf{A}\Sigma^{(p)}\right].\end{aligned} \tag{2.16}$$

**2.2. Nonstationary observation model.** In many applications, such as radio astronomy or airborne radar, the relationship between the observed and the observer is changing over time. An extension of the model presented in the previous subsection is one in which each observation is taken through a different linear transformation $\mathbf{A}_k$. In this case, one cannot aggregate all of the data into a single sample covariance, but rather the conditional expectation of the complete-data sufficient statistics must be taken on each data vector or perhaps the sample covariance for a subset of the data vectors.

The extended model for the data is described by

$$\mathbf{y}_k = \mathbf{A}_k x_k, \tag{2.17}$$

and the incomplete-data distribution is thus given by

$$\mathbf{y}_k \sim CN(0, \mathbf{R}_k), \tag{2.18}$$

where

$$\mathbf{R}_k = \mathbf{A}_k \Sigma \mathbf{A}_k^{\mathrm{H}}. \tag{2.19}$$

We now have an estimation problem in which one must estimate not just a single covariance $\mathbf{R}$ but rather an entire sequence $\mathbf{R}_1 \dots \mathbf{R}_K$. The "thread" that ties all of the $\mathbf{R}_k$ together is the desired spectrum $\Sigma$, which is considered constant over time.

The EM algorithm for this estimation problem requires that we determine the conditional expectation of the squared magnitudes of the components of $\mathbf{x}_k$, as before. The results are then averaged together in the M-step of the algorithm.

Denote the estimated covariance for data vector $\mathbf{y}_k$ at EM iteration $p$ as $\mathbf{R}_{\mathbf{yy}}(p,k)$. Given $\Sigma^{(p)}$ and $\mathbf{y}_k$, the conditional mean of $\mathbf{x}_k$ is

$$\begin{aligned}\mathrm{E}\{\mathbf{x}_k|\mathbf{y}_k, \Sigma^{(p)}\} &= \mathbf{R}_{\mathbf{xy}}(p,k)\mathbf{R}_{\mathbf{yy}}^{-1}(p,k)\mathbf{y}_k \\ &= \Sigma^{(p)}\mathbf{A}_k^{\mathrm{H}}\mathbf{R}_{\mathbf{yy}}^{-1}(p,k)\mathbf{y}_k.\end{aligned} \tag{2.20}$$

The conditional covariance of $\mathbf{x}_k$ is given by

$$\begin{aligned}cov\{\mathbf{x}_k|\mathbf{y}_k, \Sigma^{(p)}\} &= \mathbf{R}_{\mathbf{xx}}(p,k) - \mathbf{R}_{\mathbf{xy}}(p,k)\mathbf{R}_{\mathbf{yy}}^{-1}(p,k)\mathbf{R}_{\mathbf{yx}}(p,k) \\ &= \Sigma - \Sigma^{(p)}\mathbf{A}_k^{\mathrm{H}}\mathbf{R}_{\mathbf{yy}}^{-1}(p,k)\mathbf{A}_k\Sigma^{(p)}.\end{aligned} \tag{2.21}$$

The desired conditional variances are the diagonal elements of (2.21). Averaging the conditional expectations of the squared magnitudes of the $\mathbf{x}_k$ components, we arrive at the EM algorithm step

$$\begin{aligned}\Sigma^{(p+1)} = \Sigma^{(p)} &+ \frac{1}{K}\sum_{k=1}^{K}\operatorname{diag}\left[\Sigma^{(p)}\mathbf{A}_k^{\mathrm{H}}\mathbf{R}^{-1}(p,k)\mathbf{S}_k\mathbf{R}^{-1}(p,k)\mathbf{A}_k\Sigma^{(p)}\right] \\ &- \frac{1}{K}\sum_{k=1}^{K}\operatorname{diag}\left[\Sigma^{(p)}\mathbf{A}_k^{\mathrm{H}}\mathbf{R}^{-1}(p,k)\mathbf{A}_k\Sigma^{(p)}\right],\end{aligned} \tag{2.22}$$

where

$$\mathbf{R}(p,k) = \mathbf{A}_k \Sigma^{(p)} \mathbf{A}_k^{\mathrm{H}}. \tag{2.23}$$

Here $\mathbf{S}_k$ is either $\mathbf{y}_k \mathbf{y}_k^{\mathrm{H}}$ or the sample covariance for the $k$th observation interval, should that include more than one observation for the same $\mathbf{A}_k$.

**2.3. Additive noise model.** In some applications, such as radar imaging, the data are observed in the presence of additive white instrument or background noise. In this case, the data model is (reverting to the stationary observation case)

$$\mathbf{y}_k = \mathbf{A}\mathbf{x}_k + \mathbf{n}_k, \tag{2.24}$$

where $\mathbf{n}_k$ is $CN(0, \epsilon\mathbf{I})$ and independent of $\mathbf{x}_k$. It follows that

$$\mathbf{y}_k \sim CN(0, \mathbf{A}\Sigma\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I}). \tag{2.25}$$

Again the basic task in the EM algorithm is to compute the conditional distribution for $\mathbf{x}_k$ given $\mathbf{y}_k$ and $\Sigma$. The conditional mean is

$$\begin{aligned} \mathrm{E}\{\mathbf{x}_k | \mathbf{y}_k, \Sigma\} &= \mathbf{R}_{\mathbf{xy}} \mathbf{R}_{\mathbf{yy}}^{-1} \mathbf{y}_k \\ &= \Sigma \mathbf{A}^{\mathrm{H}} (\mathbf{A}\Sigma\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1} \mathbf{y}_k. \end{aligned} \tag{2.26}$$

The conditional covariance is

$$\begin{aligned} cov\{\mathbf{x}_k | \mathbf{y}_k, \Sigma\} &= \mathbf{R}_{\mathbf{xx}} - \mathbf{R}_{\mathbf{xy}} \mathbf{R}_{\mathbf{yy}}^{-1} \mathbf{R}_{\mathbf{yx}} \\ &= \Sigma - \Sigma \mathbf{A}^{\mathrm{H}} (\mathbf{A}\Sigma\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1} \mathbf{A}_k \Sigma. \end{aligned} \tag{2.27}$$

Folding these into the EM algorithm as before we get

$$\begin{aligned} \Sigma^{(p+1)} = \Sigma^{(p)} &+ \mathrm{diag}\left[\Sigma^{(p)} \mathbf{A}^{\mathrm{H}} (\mathbf{A}\Sigma^{(p)}\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1} \mathbf{S} (\mathbf{A}\Sigma^{(p)}\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1} \mathbf{A}\Sigma^{(p)}\right] \\ &- \mathrm{diag}\left[\Sigma^{(p)} \mathbf{A}^{\mathrm{H}} (\mathbf{A}\Sigma^{(p)}\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1} \mathbf{A}\Sigma^{(p)}\right]. \end{aligned} \tag{2.28}$$

This result can be extended to the the nonstationary observation model in the obvious way, by averaging the complete-data sufficient statistics over the $K$ observations.

## 3. Basic computational tasks.

In the derivation above, no mention was made of the size or rank of the matrices $\mathbf{A}$ or $\mathbf{A}_k$, or even whether or not the covariances $\mathbf{R}$ or $\mathbf{R}_k$ are invertible. In fact, the results obtained above for the no-noise case are valid only in the case where $\mathbf{A}$ ($\mathbf{A}_k$) is wide and full-rank. Nevertheless, we now show that there exists a single interpretation of the E-step for the underdetermined, overdetermined, full-rank, and rank-deficient cases of the basic problem. The basic computations for the stationary and nonstationary observation cases are the same, the only difference being in the aggregation of data into sample covariance matrices under the stationary observation case. The result is slightly different for the additive noise case, so it will be treated separately.

**3.1. No-noise model.** The essential computational task at iteration $p$ is to compute the conditional distribution of the $\mathbf{x}_k$ given the incomplete data $\mathbf{y}_k$ and the current $\Sigma^{(p)}$. For notational simplicity, we will drop the superscript $p$ (iteration number) and the subscript $k$ (data index) and concern ourselves with the calculation of the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ and $\Sigma$. We assume without loss of generality that all elements of $\Sigma$ are positive. If any $\sigma(n) = 0$, then the conditional expected value of $s(n)$ is zero as well, and the $n$th column of $\mathbf{A}$ can be removed from consideration.

Let $\mathbf{A}$ be $M \times N$, with rank $r \leq \min(M, N)$. For the purposes of this derivation, it is assumed that $r$ is known and that the $\min(M, N) - r$ singular values of $\mathbf{A}$ are exactly 0. In section 4, the results will be applied to the case more commonly encountered in practice, in which $r$ is the "numerical rank" determined by comparing the smallest singular values to a small but nonzero threshold.

Define the matrix $\mathbf{B}$ by

$$\mathbf{B} = \mathbf{A}\Sigma^{\frac{1}{2}}. \tag{3.1}$$

Let the SVD of $\mathbf{B}$ be given by

$$\mathbf{A}\Sigma^{\frac{1}{2}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{H}}. \tag{3.2}$$

This is the "full-size" SVD, where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices of size $M \times M$ and $N \times N$, respectively, and $\mathbf{D}$ is a diagonal rectangular $M \times N$ matrix, with

$$\begin{aligned} \mathbf{D}(i,i) &= d_i, \quad i = 1 \ldots r, \\ &= 0, \quad \text{otherwise.} \end{aligned} \tag{3.3}$$

The $d_i$ are the singular values of $\mathbf{B}$, and the columns of $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors, respectively, of $\mathbf{B}$.

Define

$$\mathbf{z} = \mathbf{U}^{\mathrm{H}}\mathbf{y} \tag{3.4}$$

and

$$\mathbf{w} = \Sigma^{-\frac{1}{2}}\mathbf{x}. \tag{3.5}$$

Both of these mappings are one-to-one; hence, observation of $\mathbf{z}$ is equivalent to observation of $\mathbf{y}$, and the conditional distribution of $\mathbf{x}$ can be determined from the conditional distribution of $\mathbf{w}$. It suffices to determine the conditional distribution of $\mathbf{w}$ given $\mathbf{z}$. These two vectors are related by the linear transformation

$$\mathbf{z} = \mathbf{D}\mathbf{V}^{\mathrm{H}}\mathbf{w}. \tag{3.6}$$

The unconditional distribution for $\mathbf{w}$ is

$$\mathbf{w} \sim CN(0, \mathbf{I}_N). \tag{3.7}$$

Now define $\tilde{\mathbf{w}} = \mathbf{V}^{\mathrm{H}}\mathbf{w}$. $\mathbf{V}$ is a unitary, and hence invertible, transformation, and

$$\tilde{\mathbf{w}} \sim CN(0, \mathbf{I}_N). \tag{3.8}$$

Since $\mathbf{Z} = \mathbf{D}\tilde{\mathbf{w}}$ there are $r$ observed components of $\tilde{\mathbf{w}}$, namely, $\tilde{w}(1) \ldots \tilde{w}(r)$, and the remaining $N - r$ entries of $\tilde{\mathbf{w}}$ are unobserved. Furthermore, these other $N - r$ entries are uncorrelated with the first $r$, so their distribution is not changed under conditioning. Thus we have

$$\begin{aligned} \mathrm{E}\{\tilde{w}(n)|\mathbf{z}\} &= \frac{z(n)}{d_n}, \quad n = 1 \ldots r, \\ &= 0, \qquad n = r+1 \ldots N \end{aligned} \tag{3.9}$$

and

$$cov\{\tilde{\mathbf{w}}|\mathbf{z}\} = \begin{bmatrix} \mathbf{0}_{r\times r} & \mathbf{0}_{(N-r)\times r} \\ \mathbf{0}_{r\times (N-r)} & \mathbf{I}_{(N-r)\times(N-r)} \end{bmatrix}. \tag{3.10}$$

The conditional distribution for $\mathbf{w}$ given $\mathbf{y}$ follows immediately from the conditional distribution for $\tilde{\mathbf{w}}$ given $\mathbf{z}$. Using $\mathbf{w} = \mathbf{V}\tilde{\mathbf{w}}$ and $\mathbf{z} = \mathbf{U}^{\mathrm{H}}\mathbf{y}$ we have

$$\mathrm{E}\{\mathbf{w}|\mathbf{y}\} = \mathbf{V}\,\mathrm{diag}\left[d_1^{-1}\dots d_r^{-1} 0 \dots 0\right] \mathbf{U}^{\mathrm{H}}\mathbf{y} \tag{3.11}$$

and

$$cov\{\mathbf{w}|\mathbf{y}\} = \mathbf{V}_2\mathbf{V}_2^{\mathrm{H}}. \tag{3.12}$$

In (3.12), $\mathbf{V}_2$ comprises columns $r+1\dots N$ of the unitary matrix $\mathbf{V}$ and spans the orthogonal complement of the row space of $\mathbf{B} = \mathbf{A}\Sigma^{\frac{1}{2}}$.

The conditional mean in (3.11) is the [illegible] (MNLS) solution to the rectangular system of equations

$$\mathbf{y} = \mathbf{B}\mathbf{w}, \tag{3.13}$$

which we denote as $\hat{\mathbf{w}}_{MNLS}$. As the name implies, there are two aspects of the MNLS solution: the least-squares part and the minimum-norm part. The solution is [illegible] because the reconstruction $\mathbf{B}\hat{\mathbf{w}}_{MNLS}$ is as close as possible to $\mathbf{y}$ and lies in the column space of $\mathbf{B}$. It is [illegible] because $\hat{\mathbf{w}}_{MNLS}$ is orthogonal to the row space of $\mathbf{B}$. The matrix that multiplies $\mathbf{y}$ in (3.11) is called the [illegible] of the matrix $\mathbf{B}$ [18, 19].

The conditional distribution of $\mathbf{x}$ now follows immediately from (3.11)–(3.12) and (3.5). It is given by

$$E\{\mathbf{x}|\mathbf{y}\} = \Sigma^{\frac{1}{2}}\hat{\mathbf{w}}_{MNLS} \tag{3.14}$$

and

$$cov\{\mathbf{x}|\mathbf{y}\} = \Sigma^{\frac{1}{2}}\mathbf{V}_2\mathbf{V}_2^{\mathrm{H}}\Sigma^{\frac{1}{2}}. \tag{3.15}$$

The EM algorithm requires the computation of the squared magnitudes of the components of (3.14) and the diagonal elements of (3.15). If there are multiple observations $\mathbf{y}_1\dots\mathbf{y}_K$, then either the calculation in (3.14) is repeated $K$ times or an operation can be carried out on the sample covariance $\mathbf{S}$. The diagonal elements of (3.15) are found by summing the squared magnitudes of the rows of $\Sigma\mathbf{V}_2$.

When $\mathbf{A}$ is rectangular with $M \leq N$ and full-rank, then the required matrix inverses exist, and the iteration obtained here coincides with the usual EM algorithm. Note that, in this case, (2.16) can be written

$$\begin{aligned} \Sigma^{(p+1)} = \mathrm{diag}\Big[&\Sigma^{(p)\frac{1}{2}}\mathbf{B}^{\mathrm{H}}(\mathbf{B}\mathbf{B}^{H})^{-1}\mathbf{S}(\mathbf{B}\mathbf{B}^{H})^{-1}\mathbf{B}\Sigma^{(p)\frac{1}{2}} \\ &+\Sigma^{(p)} - \Sigma^{(p)\frac{1}{2}}\mathbf{B}^{H}(\mathbf{B}\mathbf{B}^{\mathrm{H}})^{-1}\mathbf{B}\Sigma^{(p)\frac{1}{2}}\Big]. \end{aligned} \tag{3.16}$$

In summary, each iteration of the EM algorithm in the no-noise case requires (a) the MNLS solution to the rectangular system of equations

$$\mathbf{y} = (\mathbf{A}\Sigma^{\frac{1}{2}})\mathbf{w} \tag{3.17}$$

and (b) an orthogonal basis for the row space of $\mathbf{A}\Sigma^{\frac{1}{2}}$.

**3.2. Additive noise case.** In the presence of additive noise, the basic computation changes somewhat. In a certain sense, the problem is better-posed in this case since the matrix inverses in the EM iterations of section 2.3 exist, independent of $M$, $N$, and $r$.

Again the basic computational task is the calculation of the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ and $\Sigma$. Define $L = \min(M, N)$ and write the "economy-size" SVD of $\mathbf{B} = \mathbf{A}\Sigma^{\frac{1}{2}}$ as

$$\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{H}}, \tag{3.18}$$

where $\mathbf{U}$ is $M \times L$, $\mathbf{D}$ is $L \times L$, and $\mathbf{V}$ is $N \times L$. The conditional mean of $\mathbf{x}$ is

$$\mathrm{E}\{\mathbf{x}|\mathbf{y};\Sigma\} = \Sigma^{\frac{1}{2}}(\mathbf{V}\mathbf{D}\mathbf{U}^{\mathrm{H}})(\mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1}\mathbf{y}. \tag{3.19}$$

The matrix inverse in (3.19) exists independent of $M$, $N$, and $r$. We have that

$$\begin{aligned} \mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{I} &= \mathbf{U}(\mathbf{D}^2 + \epsilon\mathbf{I})\mathbf{U}^{\mathrm{H}} \quad (M < N) && \text{(3.20a)} \\ &= \mathbf{U}(\mathbf{D}^2 + \epsilon\mathbf{I})\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{U}_2\mathbf{U}_2^{\mathrm{H}} \quad (M > N), && \text{(3.20b)} \end{aligned}$$

where, in the $M > N$ case, $\mathbf{U}_2$ is the orthonormal matrix whose columns span the orthogonal complement of the column range of $\mathbf{U}$. In either case, we have

$$\begin{aligned} \mathrm{E}\{\mathbf{x}|\mathbf{y};\Sigma\} &= \Sigma^{\frac{1}{2}}\mathbf{V}\mathbf{D}\mathbf{U}^{\mathrm{H}}\mathbf{U}(\mathbf{D}^2 + \epsilon\mathbf{I})^{-1}\mathbf{U}^{\mathrm{H}}\mathbf{y} \\ &= \Sigma^{\frac{1}{2}}\mathbf{V}\begin{bmatrix} \frac{d_1}{d_1^2 + \epsilon} & & \\ & \ddots & \\ & & \frac{d_L}{d_L^2 + \epsilon} \end{bmatrix}\mathbf{U}^{\mathrm{H}}\mathbf{y}. \end{aligned} \tag{3.21}$$

Note that when $r < L$, there are no singularities or discontinuities; $L - r$ elements of the diagonal matrix in (3.21) go to 0 continuously as $d_i \to 0$. For this reason we define the matrix in (3.21) as a [illegible], in contrast to the Moore–Penrose pseudoinverse, which requires a hard decision as to whether a singular value is above or below some tolerance.

Using the same economy-size SVD, the conditional covariance of $\mathbf{x}$ is

$$\begin{aligned} cov\{\mathbf{x}|\mathbf{y};\Sigma\} &= \Sigma^{\frac{1}{2}}\mathbf{V}\mathbf{D}\mathbf{U}^{\mathrm{H}}\mathbf{U}(\mathbf{D}^2 + \epsilon\mathbf{I})^{-1}\mathbf{U}\mathbf{U}^{\mathrm{H}}\mathbf{D}\mathbf{V}^{\mathrm{H}}\Sigma^{\frac{1}{2}} \\ &= \Sigma^{\frac{1}{2}}\mathbf{V}\begin{bmatrix} \frac{d_1^2}{d_1^2 + \epsilon} & & \\ & \ddots & \\ & & \frac{d_L^2}{d_L^2 + \epsilon} \end{bmatrix}\mathbf{V}^{\mathrm{H}}\Sigma^{\frac{1}{2}}. \end{aligned} \tag{3.22}$$

Since only the diagonal elements of (3.22) are required, it suffices to compute $\mathbf{V}$ first, then to compute the weighted inner products

$$\sum_{j=1}^{L} \alpha_j |v(i,j)|^2, \tag{3.23}$$

with

$$\alpha_j = \frac{d_j^2}{d_j^2 + \epsilon}, \tag{3.24}$$

followed by pointwise multiplication by the elements of $\Sigma$.

**4. Computational alternatives.** As shown in section 2.2 above, the primary computational task at each iteration of the EM algorithm is the calculation of some sort of pseudoinverse of the system of equations

$$\mathbf{y} = \mathbf{B}\mathbf{w}, \tag{4.1}$$

where $\mathbf{B} = \mathbf{A}\Sigma^{\frac{1}{2}}$. Any routine that does this must be general-purpose and numerically stable, meaning that it should handle any combination of $(M, N)$ (the size of $\mathbf{A}$) and rank $r$. In the no-noise case, we need the Moore–Penrose pseudoinverse (or hard-decision pseudoinverse) given by

$$\mathbf{B}_{MP}^{\#} = \mathbf{V}\mathbf{E}_{MP}\mathbf{U}^{\mathrm{H}}, \tag{4.2}$$

with

$$\begin{aligned} e_{MP}(i,i) &= \frac{1}{d_i}, \quad d_i > tol, \\ &= 0, \quad\ \ d_i < tol. \end{aligned} \tag{4.3}$$

In the additive-noise case, the solution is given by the soft-decision pseudoinverse

$$\mathbf{B}_{SD}^{\#} = \mathbf{V}\mathbf{E}_{SD}\mathbf{U}^{\mathrm{H}}, \tag{4.4}$$

with

$$e_{SD}(i,i) = \frac{d_i}{d_i^2 + \epsilon}. \tag{4.5}$$

To calculate the conditional covariance term of the EM algorithm we must also know an orthonormal basis for the row space of $\mathbf{A}\Sigma^{\frac{1}{2}}$.

**4.1. SVD-based approach.** We advocate a computational approach based on the SVD, following exactly the development in section 3. There is obviously a significant computational overhead associated with computing the SVD of $\mathbf{A}\Sigma^{\frac{1}{2}}$; however, all of the required quantities in the EM algorithm follow trivially from the SVD, as has already been shown. Furthermore, all computations are numerically stable, and all possible combinations of $M$, $N$, and $r$ are unified in a single algorithm.

The use of the soft-decision pseudoinverse could also unify the additive-noise and no-noise models. Since it can be argued that any data acquisition system has some level of additive noise, one could always use the additive-noise model with some small value of $\epsilon$. One very important advantage of the soft-decision pseudoinverse is that it is continuous with respect to small changes in the matrix $\mathbf{B}$. Since $\Sigma$ is changing through the course of the EM algorithm, the fact that the Moore–Penrose pseudoinverse is discontinuous may lead to unpredictable numerical behavior—say, for example, when the numerical rank of $\mathbf{A}\Sigma^{\frac{1}{2}}$ changes from one iteration to the next.

A second recommendation, for computational efficiency, is to replace the data matrix $\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_K]$ with the lower-triangular Cholesky factor of the sample covariance matrix when $K > M$. Note that

$$\mathbf{S} = \frac{1}{K}\mathbf{Y}\mathbf{Y}^{\mathrm{H}}. \tag{4.6}$$

When $K > M$, one can write the LQ factorization of $\mathbf{Y}$ as

$$\mathbf{Y} = \mathbf{L}\mathbf{Q}, \tag{4.7}$$

where $\mathbf{L}$ is an $M \times M$ lower-triangular matrix, and $\mathbf{Q}$ is an $M \times K$ matrix with orthonormal rows. Since $\mathbf{Q}\mathbf{Q}^{\mathrm{H}} = \mathbf{I}$, we have that

$$\mathbf{S} = \frac{1}{K}\mathbf{L}\mathbf{L}^{\mathrm{H}}, \tag{4.8}$$

or, put another way, $\mathbf{L}$ is the lower Cholesky factor of $K\mathbf{S}$. In the algorithm summary below, the matrix $\mathbf{Y}$ can represent either this triangular factor, which needs to be computed only once, or in the $K < M$ case it can represent the data matrix directly.

We summarize here the basic computational step of the EM algorithm, using the SVD and the soft-decision pseudoinverse. This description is for the stationary observation model; the extension to the nonstationary observation is to repeat the calculation below for each of the observations and average the $\Sigma$'s which result.

Algorithm 1 (SVD). [illegible] $\mathbf{A}$ [illegible] $\mathbf{Y}$ [illegible] $\epsilon$ [illegible] $\Sigma^{(p)}$ [illegible]

1. [illegible] $\mathbf{A}\Sigma^{\frac{1}{2}}$

$$\mathbf{A}(\Sigma^{(p)})^{\frac{1}{2}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{H}}. \tag{4.9}$$

2. [illegible]

$$\mathbf{T}_1 = \mathbf{V}\mathbf{E}_{SD}\mathbf{U}^{\mathrm{N}}\mathbf{Y}, \tag{4.10}$$

[illegible]

$$\mathbf{E}_{SD} = \mathrm{diag}\left[\frac{d_1}{d_1^2+\epsilon}\cdots\frac{d_M}{d_M^2+\epsilon}\right]. \tag{4.11}$$

3. [illegible]

$$\mathbf{T}_2 = \mathbf{V}. \tag{4.12}$$

4. [illegible] $n = 1 \ldots N$ [illegible]

$$\sigma_n^{(p+1)} = \sigma_n^p\left[1 + \frac{1}{K}\sum_{j=1}^{min(K,M)}|t_1(n,j)|^2 - \sum_{j=1}^{M}\alpha_j|t_2(n,j)|^2\right], \tag{4.13}$$

[illegible]

$$\alpha_j = \frac{d_j^2}{d_j^2+\epsilon}. \tag{4.14}$$

**4.2. LQ-based approach.** The SVD-based algorithm described above has desirable numerical properties and works for all combinations of $M$, $N$, and $r$ but does suffer from the overhead of computing the SVD itself. We describe now an alternative method which is applicable in the commonly encountered situation in which (a) no noise is included in the model, (b) $M < N$, and (c) $\mathbf{A}\Sigma^{\frac{1}{2}}$ is full-rank. Condition (c) is guaranteed as long as $\mathbf{A}$ is full-rank and $\Sigma$ is all-positive, or, alternatively, if $\mathbf{A}$ has the property that any collection of $M$ columns is linearly independent, and at least $M$ diagonal elements of $\Sigma$ are nonzero.

Under the conditions stated above, there exists a decomposition of $\mathbf{B} = \mathbf{A}\Sigma^{\frac{1}{2}}$, called the LQ decomposition, given by

$$\mathbf{B} = \mathbf{L}\mathbf{Q}, \tag{4.15}$$

where $\mathbf{L}$ is an $M \times M$ lower-triangular matrix, and $\mathbf{Q}$ is an $M \times N$ matrix with orthonormal rows. (This is the transpose of the QR decomposition of $\mathbf{B}^{\mathrm{H}}$.) Calculation of the LQ decompositions requires $O(M^2N)$ flops.

Given the LQ decomposition of $\mathbf{B}$, the conditional mean of $\mathbf{x}$ given $\mathbf{y}$ and $\Sigma$ is

$$\begin{aligned} E\{\mathbf{x}|\mathbf{y}\} &= \Sigma^{\frac{1}{2}}\mathbf{B}^{\mathrm{H}}(\mathbf{B}\mathbf{B}^{\mathrm{H}})^{-1}\mathbf{y} \\ &= \Sigma^{\frac{1}{2}}\mathbf{Q}^{\mathrm{H}}\mathbf{L}^{\mathrm{H}}(\mathbf{L}\mathbf{Q}\mathbf{Q}^{\mathrm{H}}\mathbf{L}^{\mathrm{H}})^{-1}\mathbf{y} \\ &= \Sigma^{\frac{1}{2}}\mathbf{Q}^{\mathrm{H}}\mathbf{L}^{-1}\mathbf{y}. \end{aligned} \tag{4.16}$$

The triangular backsolve $\mathbf{L}^{-1}\mathbf{y}$ requires $O(M^2)$ flops, and multiplication by $\Sigma^{\frac{1}{2}}\mathbf{Q}^{\mathrm{H}}$ requires $O(NM)$ flops. For $K$ vectors $\mathbf{y}_1 \dots \mathbf{y}_K$, or some other factorization of the sample covariance matrix with $K$ columns, the above computational estimates are multiplied by a factor of $K$.

The conditional covariance of $\mathbf{x}$ given $\mathbf{y}$ is

$$\begin{aligned} cov\{\mathbf{x}|\mathbf{y}\} &= \Sigma - \Sigma^{\frac{1}{2}}\mathbf{B}^{\mathrm{H}}(\mathbf{B}\mathbf{B}^{\mathrm{H}})^{-1}\mathbf{B}\Sigma^{\frac{1}{2}} \\ &= \Sigma^{\frac{1}{2}}(\mathbf{I} - \mathbf{Q}^{\mathrm{H}}\mathbf{Q})\Sigma^{\frac{1}{2}}. \end{aligned} \tag{4.17}$$

The diagonal elements of (4.17) are found by computing the column sums $\sum_{i=1}^{M} |q_{ij}|^2$ and then weighting by $\sigma_j$. This requires $O(MN)$ operations.

The summary of the LQ algorithm is as given below.

ALGORITHM 2 (LQ). [illegible] $\mathbf{A}$ [illegible] $\mathbf{Y}$ [illegible] $\Sigma^{(p)}$

1. [illegible] $\mathbf{A}\Sigma^{\frac{1}{2}}$

$$\mathbf{A}(\Sigma^{(p)})^{\frac{1}{2}} = \mathbf{L}\mathbf{Q}. \tag{4.18}$$

2. [illegible]

$$\mathbf{T}_1 = \mathbf{Q}^{\mathrm{H}}\mathbf{L}^{-1}\mathbf{Y}. \tag{4.19}$$

3. [illegible]

$$\mathbf{T}_2 = \mathbf{Q}^{\mathrm{H}}. \tag{4.20}$$

4. [illegible] $n = 1 \dots N$ [illegible]

$$\sigma_n^{(p+1)} = \sigma_n^p \left[ 1 + \frac{1}{K}\sum_{j=1}^{K} |t_1(n,j)|^2 - \sum_{j=1}^{M} |t_2(n,j)|^2 \right] \tag{4.21}$$

Given the respective decompositions, both the LQ and the SVD methods require a similar computational load. An informal study of the QR and SV decompositions indicates that, while both are $O(NM^2)$ algorithms, the SVD requires about 3 times the computation of the QRD. This suggests that the LQ approach may be an attractive alternative as long as the conditions set forth at the beginning of this section are satisfied. The primary benefit of using the SVD is the "peace of mind" that comes with knowing that no special efforts must be made to ensure invertibility of matrices involved in the calculations. We also point out that, as $K$ (the number of columns in the factor of the sample covariance matrix) approaches $M$, the calculations in the EM algorithm after the decomposition become comparable with the decomposition itself, and thus the relative advantage of the LQ over the SVD method begins to disappear.

**4.3. Calculation of related quantities.** The calculations described above are central not only to the structured covariance EM algorithm but to closely related quantities in the maximum-likelihood estimation problem, namely, the log-likelihood and the gradient of the log-likelihood. The log-likelihood is given by

$$l(\mathbf{R};\mathbf{S}) = -\log\det\mathbf{R} - \mathrm{tr}\mathbf{R}^{-1}\mathbf{S}, \tag{4.22}$$

and the gradient of the log-likelihood, with respect to $\Sigma$ and expressed as a diagonal matrix, is

$$\mathbf{G}(\Sigma;\mathbf{S}) = \mathrm{diag}\left[\mathbf{A}^{\mathrm{H}}(\mathbf{R}^{-1}\mathbf{S}\mathbf{R}^{-1} - \mathbf{R}^{-1})\mathbf{A}^{\mathrm{H}}\right]. \tag{4.23}$$

We use here only the additive noise model for which $\mathbf{R} = \mathbf{A}\Sigma\mathbf{A}^{\mathrm{H}} + \epsilon\mathbf{I}$.

Let the SVD of $\mathbf{A}\Sigma^{\frac{1}{2}}$ be $\mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{H}}$ as before. Then the expression for $\mathbf{R}$ is

$$\mathbf{R} = \mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{I}. \tag{4.24}$$

It follows that the first term in the log-likelihood is

$$\log\det\mathbf{R} = \sum_{i=1}^{M}\log(d_i^2+\epsilon) \quad (M \le N) \tag{4.25a}$$

$$= \sum_{i=1}^{M}\log(d_i^2+\epsilon) + (M-N)\log\epsilon \quad (M > N). \tag{4.25b}$$

The second term in the log-likelihood is

$$\mathrm{tr}\mathbf{R}^{-1}\mathbf{S} = \mathrm{tr}\mathbf{Y}^{\mathrm{H}}(\mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1}\mathbf{Y} \tag{4.26a}$$

$$= \sum_{i=1}^{M}\frac{1}{d_i^2+\epsilon}\sum_{j=1}^{K}|(\mathbf{U}^{\mathrm{H}}\mathbf{Y})(i,j)|^2 \quad (M \le N) \tag{4.26b}$$

$$= \sum_{i=1}^{M}\frac{1}{d_i^2+\epsilon}\sum_{j=1}^{K}|(\mathbf{U}^{\mathrm{H}}\mathbf{Y})(i,j)|^2 + \frac{1}{\epsilon}\sum_{i=1}^{M-N}\sum_{j=1}^{K}|(\mathbf{U}_2^{\mathrm{H}}\mathbf{Y})(i,j)|^2 \quad (M > N), \tag{4.26c}$$

where, in the second $M > N$ case, $\mathbf{U}_2$ is an orthonormal basis spanning the orthogonal complement of the range of $\mathbf{U}$.

Calculation of the gradient is easy as long as none of the $\sigma_i$ are equal to 0. The EM iteration itself can be written as

$$\Sigma^{(p+1)} = \Sigma^{(p)} + (\Sigma^{(p)})^2\mathbf{G}^{(p)}, \tag{4.27}$$

where

$$\mathbf{G}^{(p)} = \mathbf{G}(\Sigma^{(p)};\mathbf{S}) \tag{4.28}$$

as given in (4.23). If $\mathbf{T}^{(p)}$ is defined as the diagonal matrix of quantities inside the square brackets in (4.13), then

$$\Sigma^{(p+1)} = \Sigma^{(p)} + \Sigma^{(p)}\mathbf{T}^{(p)}. \tag{4.29}$$

It follows that

$$\mathbf{G}^{(p)} = (\Sigma^{(p)})^{-1}\mathbf{T}^{(p)}. \tag{4.30}$$

The difficulty with the expression in (4.30) is that some of the $\sigma_i$ may be 0. The effect of this on $\mathbf{T}$, by the SVD of $\mathbf{A}\Sigma^{\frac{1}{2}}$, is that the corresponding $t_{ii}$ is also 0, and thus the correct value of the gradient is indeterminate from (4.30).

When the possibility exists for some of the $\sigma_j$ to be equal to 0, then an approach to computing the gradient that relies on direct calculation with $\mathbf{A}$ is required. Given the SVD of $\mathbf{A}\Sigma^{\frac{1}{2}} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathrm{H}}$ as before, the gradient can be written

$$\begin{aligned}\mathbf{G}(\Sigma;\mathbf{S}) = \operatorname{diag}&\left[\mathbf{A}^{\mathrm{H}}(\mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1}\mathbf{Y}\mathbf{Y}^{\mathrm{H}}(\mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon\mathbf{I})^{-1}\mathbf{A}\right] \\ &- \operatorname{diag}\left[\mathbf{A}^{\mathrm{H}}(\mathbf{U}\mathbf{D}^2\mathbf{U}^{\mathrm{H}} + \epsilon I)^{-1}\mathbf{A}\right].\end{aligned} \tag{4.31}$$

For the case $M < N$, define $\mathbf{B} = \mathbf{U}^{\mathrm{H}}\mathbf{A}$ and $\mathbf{Z} = \mathbf{U}^{\mathrm{H}}\mathbf{Y}$. Further define

$$\mathbf{T}_1 = \mathbf{B}^{\mathrm{H}}(\mathbf{D}^2 + \epsilon\mathbf{I})^{-1}\mathbf{Z} \tag{4.32}$$

and

$$\mathbf{T}_2 = (\mathbf{D}^2 + \epsilon\mathbf{I})^{-\frac{1}{2}}\mathbf{B}. \tag{4.33}$$

Then the $j$th term of the gradient (the $j$th diagonal element of $\mathbf{G}$) is given by

$$g_j = \sum_{i=1}^{M} |t_1(i,j)|^2 - \sum_{i=1}^{M} |t_2(i,j)|^2. \tag{4.34}$$

Some extra calculations are required for the $M > N$ case. Define $\mathbf{B}_2 = \mathbf{U}_2^{\mathrm{H}}\mathbf{A}$ and $\mathbf{Z}_2 = \mathbf{U}_2\mathbf{Y}$, where $\mathrm{U}_2$ is any orthogonal matrix spanning the orthogonal complement of $\mathbf{U}$. Then the definitions of $\mathbf{T}_1$ and $\mathbf{T}_2$ change to

$$\mathbf{T}_1 = \mathbf{B}^{\mathrm{H}}(\mathbf{D}^2 + \epsilon\mathbf{I})^{-1}\mathbf{Z} + \epsilon^{-1}\mathbf{B}_2^{\mathrm{H}}\mathbf{Z}_2 \tag{4.35}$$

and

$$\mathbf{T}_2 = \begin{bmatrix} (\mathbf{D}^2 + \epsilon\mathbf{I})^{-\frac{1}{2}}\mathbf{B} \\ \epsilon^{-\frac{1}{2}}\mathbf{B}_2 \end{bmatrix}. \tag{4.36}$$

With these new definitions for $\mathbf{T}_1$ and $\mathbf{T}_2$, the expression for the gradient given in (4.34) remains the same. If special care is taken to ensure that the range of $\mathbf{A}$ and the range of $\mathbf{U}$ are the same, even when some values of $\sigma_j$ are zero (which causes $\mathbf{A}\Sigma^{\frac{1}{2}}$ to be rank-deficient), then $\mathbf{B}_2 = 0$, and the modifications to $\mathbf{T}_1$ and $\mathbf{T}_2$ for the $M > N$ case become unnecessary.

**5. Conclusion.** Numerically stable methods for the computations required in the EM algorithm for maximum-likelihood structured covariance estimation have been presented. The basic computational task at each iteration is the calculation of a pseudoinverse of a certain linear system of equations, which is either a hard-decision or Moore–Penrose inverse in the no-noise case or a soft-decision pseudoinverse in the additive-noise case. An approach to computing this pseudoinverse that can handle all combinations of dimension and rank in this system of equations, based on the SVD, was proposed. An alternative method based on the LQ decomposition, which is applicable in the case where $\mathbf{A}$ is wide and full-rank, was also proposed. Finally, it was shown how the intermediate calculations required in the EM algorithm can be used to calculate the log-likelihood and the gradient of the log-likelihood for this maximum-likelihood structured covariance estimation problem.

## REFERENCES

[1] L. Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1987.

[2] S. Kay, *Modern Spectral Estimation, Theory and Application*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[3] S. Haykin, ed., *Advances in Spectrum Analysis and Array Processing*, Vol. 1, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[4] S. Haykin, ed., *Advances in Spectrum Analysis and Array Processing*, Vol. 2, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[5] P. Stoica and R. Moses, *Introduction to Spectral Analysis*, Prentice-Hall, Upper Saddle River, NJ, 1997.

[6] J. Burg, D. Luenberger, and D. Wenger, *Estimation of structured covariance matrices*, Proc. IEEE, 70 (1982), pp. 963–974.

[7] M. Miller and D. Snyder, *The role of likelihood and entropy in incomplete-data problems: Application to estimating point-processing intensities and Toeplitz constrained covariances*, Proc. IEEE, 75 (1987), pp. 892–907.

[8] D. Fuhrmann and M. Miller, *On the existence of positive-definite maximum-likelihood estimates of structured covariance matrices*, IEEE Trans. Inform. Theory, 34 (1988), pp. 722–729.

[9] T. Barton and D. Fuhrmann, *Estimation of block-Toeplitz covariances*, in Proceedings of the 24th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, 1990, pp. 779–783.

[10] T. Barton and S. Smith, *Structured covariance estimation for space-time adaptive processing*, in Proceedings of the IEEE ICASSP97, Munich, Germany, 1997, pp. 3493–3496.

[11] F. Robey and D. Fuhrmann, *Structured covariance and signal estimation, adaptive beamforming and detection*, in Proceedings of the 1990 Conference on Information Science and Systems, Princeton University, Princeton, NJ, 1990, pp. 836–840.

[12] A. Lanterman, *Statistical imaging in radio astronomy via an expectation-maximization algorithm for structured covariance estimation*, in Statistical Methods in Imaging: Medicine, Optics, and Communication, J. O'Sullivan, ed., Springer-Verlag, to appear (preprint available at http://users.ece.gatech.edu/~lanterma/adl-monographs.html).

[13] D. Rieken, D. Fuhrmann, and A. Lanterman, *Spatial spectrum estimation for time-varying arrays using the EM algorithm*, in Proceedings of the 38th Allerton Conference on Communications, Control, and Computing, University of Illinois, 2000, pp. 648–657.

[14] D. Snyder, J. O'Sullivan, and M. Miller, *The use of maximum likelihood estimation for forming images of diffuse radar targets from delay-Doppler data*, IEEE Trans. Inform. Theory, 35 (1989), pp. 536–548.

[15] P. Moulin, J. O'Sullivan, and D. Snyder, *A method of sieves for multiresolution spectrum estimation and radar imaging*, IEEE Trans. Inform. Theory, 38 (1992), pp. 801–813.

[16] D. Fuhrmann and L. Boggio, *Radar imaging from multiple viewpoints and multiple noncoherent data sets*, in Proceedings of the 2004 Conference on Information Science and Systems, Princeton University, Princeton, NJ, 2004, pp. 1093–1098.

[17] A. Dempster, N. Laird, and D. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc. Ser. B, 39 (1977), pp. 1–38.

[18] C. Lawson and R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[19] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 870–894

# A MULTIGRID METHOD TO SOLVE LARGE SCALE SYLVESTER EQUATIONS*

LARS GRASEDYCK† AND WOLFGANG HACKBUSCH†

**Abstract.** We consider the Sylvester equation $AX - XB + C = 0$, where the matrix $C \in \mathbb{R}^{n\times m}$ is of low rank and the spectra of $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times m}$ are separated by a line. The solution $X$ can be approximated in a data-sparse format, and we develop a multigrid algorithm that computes the solution in this format. For the multigrid method to work, we need a hierarchy of discretizations. Here the matrices $A$ and $B$ each stem from the discretization of a partial differential operator of elliptic type. The algorithm is of complexity $\mathcal{O}(n+m)$, or, more precisely, if the solution can be represented with $(n+m)k$ data ($k \sim \log(n+m)$), then the complexity of the algorithm is $\mathcal{O}((n+m)k^2)$.

**Key words.** fast solver, Lyapunov equation, Riccati equation, Sylvester equation, control problem, low rank approximation, multigrid method

**AMS subject classifications.** 65F05, 65F30, 65F50

**DOI.** 10.1137/040618102

## 1. Introduction.

In this article we consider the matrix Sylvester equation

$$AX - XB + C = 0,$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{m\times m}$, and $C \in \mathbb{R}^{n\times m}$ are given input matrices, and the sought solution is $X \in \mathbb{R}^{n\times m}$. We make two assumptions concerning the matrices $A, B, C$.

First, $A$ and $-B$ are stiffness matrices from the discretization of a linear elliptic partial differential operator. This allows for the use of a multigrid method to solve the Sylvester equation in $\mathcal{O}(nm)$ for a general matrix $C$. In the outlook we comment on the case when $A$ and $B$ are general sparse matrices.

Second, the matrix $C$ is of low rank $k_C$, i.e., given in factorized form $C = UV^T$, with matrices $U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k}$. Under these assumptions the solution $X$ can be approximated by a matrix $\tilde{X}$ of rank $k = \mathcal{O}(|\log\varepsilon|k_C)$ such that $\|X - \tilde{X}\|_2 \le \varepsilon$. The multigrid method can be adapted so that it is of linear complexity $\mathcal{O}((n+m)k^2)$ instead of quadratic complexity.

In the following section we will consider a simple model problem where the multigrid techniques are applicable without further complications. The model is an optimal control problem that leads to an algebraic matrix Riccati equation, which can be solved iteratively by Newton's method so that in each step a Lyapunov equation $A^TX + XA = C$ has to be solved. Such a Lyapunov equation is a special case of the more general Sylvester equation. We also give an example from model reduction where the Sylvester equation appears directly for the computation of cross-Gramians. In section 2 we give a short introduction to low rank arithmetics. Section 3 examines the tensor structure of a Sylvester equation, and as a special case we consider diagonal Sylvester equations in section 4. This special case is the basis for the Jacobi iteration introduced in section 5. In section 6 we derive the multigrid method and prove its convergence. Last we present numerical results for large scale matrix equations.

---

*Received by the editors November 2, 2004; accepted for publication (in revised form) by V. Mehrmann April 24, 2007; published electronically August 22, 2007.
http://www.siam.org/journals/simax/29-3/61810.html

†Department of Scientific Computing, Max Planck Institute for Mathematics in the Sciences, Inselstr. 22-26, D-04103 Leipzig, Germany (lgr@mis.mpg.de, wh@mis.mpg.de).

**1.1. Model problem.** The model problem to be introduced in this section is the (distributed) control of the two-dimensional heat equation (cf. [13] and the references therein) which is used, e.g., in optimal control problems for the selective cooling of steel [14]. The domain where the PDE is posed is the unit square. Using a uniform tensor mesh, it allows for a simple discretization. Of course, the method that we propose is in no way limited to regular grids or simple PDEs, but it simplifies both the implementation and the presentation.

**1.1.1. Continuous model.** We fix the domain $\Omega := (0,1) \times (0,1)$ and the boundary $\Gamma := \partial\Omega$. The goal is to minimize the quadratic performance index

$$J(u) := \int_0^\infty \left(y(t)^2 + u(t)^2\right) \mathrm{d}t$$

for $u \in L^2(0,\infty)$ and the output $y \in L^2(0,\infty)$ of the corresponding control system

$$\begin{array}{llll} \partial_t x(t,\xi) & = \partial^2_{\xi_1} x(t,\xi) + \partial^2_{\xi_2} x(t,\xi) + \kappa(\xi)u(t), & \xi \in \Omega, & t \in (0,\infty), \\ x(t,\xi) & = 0, & \xi \in \Gamma, & t \in (0,\infty), \\ x(0,\xi) & = x_0, & \xi \in \Omega, & \\ y(t) & := \int_\Omega \omega(\xi) x(t,\xi) d\xi, & & t \in (0,\infty). \end{array} \tag{1.1}$$

The values of $\kappa$ and $\omega$ are

$$\kappa(\xi) := \begin{cases} 1 & \xi \in (\frac{1}{2},1) \times (0,1), \\ 0 & \text{otherwise,} \end{cases} \qquad \omega(\xi) := \begin{cases} 1 & \xi \in (0,1) \times (\frac{1}{2},1), \\ 0 & \text{otherwise.} \end{cases}$$

Here we focus on a single-input–single-output system, but a generalization to multiple inputs and multiple outputs is straightforward.

We seek the optimal control $u^*$ in linear state feedback form

$$u^*(t,\cdot) = \Pi x(t,\cdot),$$

but since an analytic solution is available only for special cases, we construct a sequence of (semi)discretizations. For each discretization level $\ell = 0, 1, \dots$ an approximation $\Pi_\ell$ to the operator $\Pi$ is computed so that $\Pi_\ell \to \Pi$ [4, 13].

**1.1.2. Semidiscretization by finite differences.** The differential equation (1.1) is discretized by finite differences on a uniform mesh of $[0,1]^2$ with $n$ interior grid points $(x_i)_{i=1}^n$ and mesh width $h = (\sqrt{n}+1)^{-1}$. By $\phi_i$ we denote the piecewise linear interpolant on the mesh with $\phi_i(x_i) = 1$ and $\phi_i(x_j) = 0$ for $j \neq i$. The corresponding space-discrete system is

$$\begin{array}{lll} \partial_t x(t) & = Ax(t) + Ku(t), & t \in (0,\infty), \\ x(0) & = x_0, & \\ y(t) & := Wx(t), & t \in (0,\infty), \end{array} \tag{1.2}$$

where $A := A^{FD} \in \mathbb{R}^{n\times n}$ is the standard finite difference discretization of the 2d Laplacian, $x(t) \in \mathbb{R}^n$, $u(t), y(t) \in \mathbb{R}$, and the vectors $K := K^{FD} \in \mathbb{R}^n$ and $W \in \mathbb{R}^n$ are

$$K_i^{FD} := \kappa(x_i), \qquad W_i := \int_\Omega \omega(\xi)\phi_i(\xi)\mathrm{d}\xi. \tag{1.3}$$

The stiffness matrix $A$ is symmetric negative definite, sparse, and ill-conditioned.

**1.1.3. Semidiscretization by finite elements.** Instead of the finite difference discretization from the previous section we can as well discretize (1.1) in the weak or variational form by finite elements on a uniform mesh of $[0,1]^2$ with $n$ interior grid points $(x_i)_{i=1}^n$, mesh width $h = (\sqrt{n}+1)^{-1}$, and piecewise linear basis functions $(\phi_i)_{i=1}^n$. The corresponding space-discrete system is (1.2), where $W$ is defined as in (1.3), $K := K^{FEM} := E^{-1}K^{FD}$ for the matrix $K^{FD}$ from (1.3), and $A := E^{-1}A^{FEM}$ for the matrices

$$A_{i,j}^{FEM} := \int_\Omega \langle \nabla\phi_i(\xi), \nabla\phi_j(\xi)\rangle \mathrm{d}\xi, \qquad E_{i,j} := \int_\Omega \phi_i(\xi)\phi_j(\xi)\mathrm{d}\xi. \tag{1.4}$$

The mass matrix $E$ is symmetric positive definite, well-conditioned, and sparse. The system matrix $A = E^{-1}A^{FEM}$ has a negative spectrum and is nonsymmetric, dense, and ill-conditioned. Therefore, one avoids working with $A$ and instead uses a generalized formulation; see (1.7).

**1.1.4. Linear state feedback control.** The discrete optimal control $u$ can be realized in linear state feedback form [12]

$$u(t) = -K^T X x(t), \qquad t \in [0,\infty),$$

where $X$ is the unique solution—in the set of symmetric positive semidefinite matrices—to the algebraic matrix Riccati equation

$$A^T X + XA - XKK^T X + WW^T = 0. \tag{1.5}$$

The matrix $A$ is of size $n \times n$. The matrices $KK^T$ and $WW^T$ are of size $n \times n$ and are data-sparse in the sense that only $K$ and $W$ have to be stored, i.e., $2n$ entries.

**1.1.5. Solution of the algebraic matrix Riccati equation.** The nonlinear equation (1.5) can be solved by Newton's method [11]. The initial guess $X_0 := 0$ is sufficient to guarantee global convergence, but in the context of multilevel methods a good initial guess can also be obtained by a coarser level solution in the nested iteration. In each step $i$ of Newton's method, we have to solve a Lyapunov equation

$$A_i^T X_i + X_i A_i + C_i = 0, \tag{1.6}$$

where the matrices $A_i$ and $C_i$ are of the form

$$A_i := A - KK^T X_{i-1}, \qquad C_i := WW^T - X_{i-1}KK^T X_{i-1}.$$

For the finite difference discretization $A = A^{FD}$ the negative definite matrix $A_i$ in the $i$th step of Newton's method is data-sparse in the sense that only the sparse $n \times n$ matrix $A$, the vector $K$, and the vector $K^T X_{i-1}$ have to be stored. For $C_i$ we have to store $W$ and $X_{i-1}K$ in addition.

For the finite element discretization it is advantageous to consider the generalized Lyapunov equation.

LEMMA 1.1. [illegible] $A, K, W$ [illegible] (1.2) [illegible] (1.4) [illegible] $\widehat{X}_i$ [illegible]

$$\widehat{A}_i^T \widehat{X}_i E + E\widehat{X}_i \widehat{A}_i + \widehat{C}_i = 0, \tag{1.7}$$

[illegible] $\widehat{A}_i, \widehat{C}_i$ [illegible]

$$\widehat{A}_i := A^{FEM} - K^{FD}(K^{FD})^T \widehat{X}_{i-1}E, \quad \widehat{C}_i := WW^T - E\widehat{X}_{i-1}K^{FD}(K^{FD})^T \widehat{X}_{i-1}E.$$

[illegible] $X_i$ [illegible] (1.6) [illegible] $X_i = E\widehat{X}_i E$

By inserting $\widehat{X}_i = E^{-1}X_iE^{-1}$ we get

$$\begin{aligned}\widehat{A}_i^T\widehat{X}_iE &= (A^{FEM} - K^{FD}(K^{FD})^T\widehat{X}_{i-1}E)^TE^{-1}X_iE^{-1}E\\ &= (E^{-1}A^{FEM} - E^{-1}K^{FD}(K^{FD})^TE^{-1}E\widehat{X}_{i-1}E)^TX_i\\ &= (E^{-1}A^{FEM} - KK^TX_{i-1})^TX_i\\ &= A_i^TX_i\end{aligned}$$

and analogously for $E\widehat{X}_i\widehat{A}_i = X_iA_i$. The right-hand side fulfils

$$\widehat{C}_i = WW^T - E\widehat{X}_{i-1}K^{FD}(K^{FD})^T\widehat{X}_{i-1}E = WW^T - X_{i-1}KK^TX_{i-1} = C_i. \qquad \square$$

The matrices $\widehat{A}_i$ in the generalized Lyapunov equation (1.7) are data-sparse in the sense that $A^{FEM}$ is sparse and the matrix $K^{FD}(K^{FD})^T\widehat{X}_{i-1}M$ of rank 1.

The Lyapunov equation (1.6) is a special Sylvester equation which is of the form

$$AX - XB + C = 0 \tag{1.8}$$

for the matrices $A := A_i^T$, $B := -A_i$, and $C := C_i$. A Sylvester equation is uniquely solvable for all matrices $C$ if and only if the spectra of $A$ and $B$ are disjoint. In our setting the matrix $A_i$ is negative definite, and therefore $A < 0$ and $B > 0$ such that the existence of a unique solution is guaranteed.

In the following subsection 1.3 we determine a suitable format for an approximation to the solution $X$ of the Sylvester equation (1.8) where the matrix $C$ is of low rank.

### 1.2. Second model problem.

The model problem of this section is identical to the linear time invariant control problem (1.1) except that the governing PDE is now

$$\dot{x}(t,\xi) = \partial_{\xi_1}^2 x(t,\xi) + \partial_{\xi_2}^2 x(t,\xi) + \beta\partial_{\xi_1}x(t,\xi) + \kappa(\xi)u(t),$$

leading to a discrete system

$$\dot{x}(t) = Ax(t) + Ku(t), \quad y(t) = W^Tx(t)$$

with a nonsymmetric matrix $A$. We aim at finding a lower order system

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{K}u(t), \quad y(t) = \hat{W}^T\hat{x}(t)$$

so that $\hat{A}$ is considerably smaller than $A$ while the input-output error is bounded and the reduced system stable [2]. The reduced system can be constructed based on a low rank approximation $\tilde{X}$ of the so-called cross-Gramian $X$ which is the solution of the Sylvester equation [2]

$$AX + XA + KW^T = 0.$$

### 1.3. Structure of the solution.

In the $i$th step of Newton's method to solve the algebraic matrix Riccati equation (1.5), we have to solve a Sylvester equation (1.8) where the matrix $C$ is of rank at most

$$\operatorname{rank}(C) \le \operatorname{rank}(WW^T) + \operatorname{rank}(X_{i-1}KK^TX_{i-1}) \le 2.$$

Since the discrete system (1.2) involves a discretization error, it is reasonable to solve the Sylvester equation only up to an accuracy $\varepsilon$ of the size of the discretization error; i.e., we seek an approximation $\tilde{X}$ to the solution $X$ of (1.8) such that

$$\|X - \tilde{X}\|_2 \leq \varepsilon \|X\|_2.$$

The idea now is to choose a matrix $\tilde{X}$ that allows for a data-sparse representation.

Definition 1.2 ($R(k)$-matrix representation). [illegible] $k, n, m \in \mathbb{N}$ [illegible] $R \in \mathbb{R}^{n\times m}$ [illegible] $R(k)$ [illegible] $R(k)$ [illegible] $R$ [illegible]

$$R = UV^T, \qquad U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{m\times k}, \tag{1.9}$$

[illegible] $U, V$ [illegible]

The two factors in the representation (1.9) of an $R(k)$-matrix involve $k(n+m)$ values to be stored. The matrix-vector multiplication $y := Rx$ can be done in two steps involving the two matrix-vector products $z := V^T x$ and $y := Uz$ that consist of $\mathcal{O}(k(n+m))$ basic arithmetic operations.

The $R(k)$-matrix format is a suitable representation for matrices of rank at most $k$: Each matrix of rank at most $k$ can be written in the factorized form (1.9) by use of a (reduced) singular value decomposition, and each matrix of the form (1.9) is of rank at most $k$. The next theorem proves the existence of a low rank approximant $\tilde{X}$ to the solution $X$ of (1.8).

Theorem 1.3 (existence of a low rank approximant). [illegible] $A \in \mathbb{R}^{n\times n}$ [illegible] $B \in \mathbb{R}^{m\times m}$ [illegible] $\sigma(A)$ [illegible] $\sigma(B)$ [illegible] $C \in \mathbb{R}^{n\times m}$ [illegible] $k_C$ [illegible] $0 < \varepsilon < 1$ [illegible] $\tilde{X} \in \mathbb{R}^{n\times m}$ [illegible] $X$ [illegible] (1.8) [illegible]

$$\|X - \tilde{X}\|_2 \leq \varepsilon \|X\|_2, \tag{1.10}$$

[illegible] $\tilde{X}$ [illegible] $\operatorname{rank}(\tilde{X}) \leq k_C k_\varepsilon$ $k_\varepsilon = \mathcal{O}(\log(1/\varepsilon))$

The proof of Theorem 1.3 is given in [6] (see also [16, 1]). One should note that the rank $k_\varepsilon$ depends on the location of the spectra of $A$ and $B$. In our model problem this is $k = \mathcal{O}(\log(1/\varepsilon)\log(n))$.

**1.4. Large scale Sylvester equations.** A fixed Sylvester equation (1.8) can, e.g., be solved by the Bartels–Stewart algorithm [3], which is of complexity $\mathcal{O}(n^3)$. In the context of large scale Sylvester equations (i.e., $n > 10^5$) one is interested in reducing the complexity for a certain class of matrices $A, B, C$.

Hu and Reichel [20] propose to use Krylov subspace methods for the solution of the Sylvester equation. In each iterative step the equation is projected to a small dimension where one can use, e.g., the Bartels–Stewart algorithm as a solution. The authors do not exploit some kind of low rank structure but the fact that $A$ and $B$ allow for a fast matrix-vector multiplication. One step of their algorithm is of complexity $\mathcal{O}(nm)$, and the necessary number of iterations increases as the condition of the Sylvester equation increases.

Li and White [15] propose an iterative method for the solution of the Lyapunov equation based on the factorization of the matrix $C$ and the solution $X$. Their method is a special implementation of the classical ADI algorithm (previously proposed by Penzl [18]) and requires the solution of a shifted linear system $A - \lambda I$ in each step. The number $J$ of steps necessary to gain a good approximation $\tilde{X}$ to $X$ depends on the choice of the shifts $\lambda$. For the nonsymmetric case there is no nontrivial upper bound for $J$. The main problem is that the approximation of rank $J$ (after $J$ steps of ADI) is not necessarily close to a best approximation of rank $J$.

Penzl [17] presents a multigrid method to compute the solution $X$ to the Lyapunov equation, but he does not exploit the fact that $X$ can—at least if $C$ is of low rank—be approximated by a low rank matrix $\tilde{X}$; therefore, the complexity of one multigrid step is $\mathcal{O}(n^2)$. He also gives a convergence analysis for a simple model problem and proves that the convergence rate is bounded independently of the problem size $n$.

In this paper, we explain how one can compute a low rank approximation $\tilde{X}$ to the solution $X$ of (1.8) by use of the multigrid method. We use the usual Jacobi smoother and standard prolongation and restriction operators but extend the basic multigrid cycle by a projection step $X_i \mapsto \mathcal{T}_k(X_i)$ that ensures that the rank of the $i$th iterate $X_i$ is bounded. For a sufficiently large rank $k$ the error $\|X_i - \mathcal{T}_k(X_i)\|$ due to the projection of the iterate $X_i$ (cf. section 2) can be regarded as the standard truncation error due to limited machine precision.

Each multigrid step is of complexity $\mathcal{O}(n+m)$, and a nested iteration combined with a level independent good convergence rate guarantees that we need only $\mathcal{O}(1)$ steps to solve the equation up to the discretization error.

The convergence analysis for simple model problems turns out to be fairly trivial. The structure of the Sylvester equation allows us to carry results for linear systems $Ax = b$ over to the Sylvester equation such that the convergence rate can be bounded also for general domains and operators. The effect of the projection to low rank in the multigrid cycle can be regarded as a reduction of the machine precision. In our numerical tests the convergence rate is not deteriorated by the projection.

## 2. $\boldsymbol{R(k)}$-matrix arithmetics.

The set of $n \times m$ $R(k)$-matrices is not a linear space because the addition of two matrices of rank at most $k$ might result in a matrix of rank larger than $k$. In this sense the $R(k)$-matrix format is not suitable for iterative solution schemes for the Sylvester equation.

However, $R(k)$-matrices allow for an efficient singular value decomposition such that the projection (a best approximation) to lower rank is of complexity $\mathcal{O}(k^2(n+m))$. This projection can be used to keep the iterates in the set $R(k)$.

LEMMA 2.1 (reduced SVD, truncation). (a) [illegible] $R = UV^T \in \mathbb{R}^{n\times m}$ [illegible] $R(k)$ [illegible] $R$ [illegible] $N_{R,\mathrm{SVD}}(n,m,k) \lesssim 6k^2(n+m) + 23k^3$ [illegible]

1. [illegible] $U = Q_U R_U$ [illegible] $U$ $Q_U \in \mathbb{R}^{n\times k}, R_U \in \mathbb{R}^{k\times k}$
2. [illegible] $V = Q_V R_V$ [illegible] $V$ $Q_V \in \mathbb{R}^{m\times k}, R_V \in \mathbb{R}^{k\times k}$
3. [illegible] $R_U R_V^T = \widetilde{U}\Sigma\widetilde{V}^T$
4. [illegible] $\widehat{U} := Q_U\widetilde{U}$ [illegible] $\widehat{V} := Q_V\widetilde{V}$

[illegible] $R = \widehat{U}\Sigma\widehat{V}^T$ [illegible] [5, [illegible] 5.2.9 [illegible] 5.4.5] [illegible]

| | | |
|---|---|---|
| [illegible] $U$ | $4nk^2$ | |
| [illegible] $V$ | $4mk^2$ | |
| [illegible] $R_U R_V^T$ | | $2k^3$ |
| [illegible] $R_U R_V^T$ | | $\approx 21k^3$ |
| [illegible] $Q_U\widetilde{U}$ [illegible] $Q_V\widetilde{V}$ | $2nk^2 + 2mk^2$ | |
| [illegible] $N_{R,\mathrm{SVD}}(n,m,k) =$ | $6k^2(n+m)+$ | $23k^3$ |

TABLE 2.1

*Time in seconds for the reduced SVD of an $n \times n$ $R(k)$-matrix, $n = 1024^2$.*

| Rank | $k = 4$ | $k = 8$ | $k = 16$ | $k = 32$ | $k = 64$ | $k = 128$ |
|---|---|---|---|---|---|---|
| Time | 6.19 | 15.32 | 49.37 | 172.73 | 653.40 | 2637.5 |

(b) [illegible] $R(k)$ [illegible] $R$ [illegible] $k' \leq k$ [illegible] $R$ [illegible] $R(k')$ [illegible] $k'$ [illegible] $\widehat{U}\Sigma$ [illegible] $\widehat{V}$ [illegible] $R$ [illegible] $k'$ [illegible]

$$\mathcal{T}_{k'}. \tag{2.1}$$

[illegible] $k' \geq k$ [illegible] $\mathcal{T}_{k'}$ [illegible] $R(k)$ [illegible] (1.9) [illegible] $U, V$ [illegible] $k' - k$ [illegible]

We remark that the truncation in part (b) becomes nonunique when the $k'$th and $(k'+1)$st singular values are equal.

LEMMA 2.2 (spectral and Frobenius norm). [illegible] $n \times m$ $R(k)$ [illegible] $R$ [illegible] 2.1a [illegible] $N_{R,\|\cdot\|}(n, m, k) \lesssim 4k^2(n + m) + 23k^3$

[illegible] The norms can be obtained from the singular values; i.e., steps 1–3 from Lemma 2.1a are to be performed. □

[illegible] 2.3 (complexity of the truncation in practice). We implement the truncation procedure of Lemma 2.1 on a SUN ULTRASPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate by use of the LAPACK subroutines `dgeqrf` and `dgesvd` for the QR-factorization and singular value decomposition of full matrices. The $1024^2 \times 1024^2$ matrix $R$ of rank $k$ is given in $R(k)$-matrix representation and has random entries in the factors $U, V$. We truncate $R$ down to rank $k/2$. The time in seconds to compute the result is given in Table 2.1.

**3. Tensor structure of the Sylvester equation.** In order to formulate and analyze the iterative solutions for the Sylvester equation, we need to reformulate the matrix equation in terms of a standard linear system of equations. For notational purposes we also introduce the Kronecker product formulation.

**3.1. Algebraic structure.** The Sylvester equation (1.8) can be written (for each entry $(i, j)$) in the form

$$\sum_{\nu=1}^{n} A_{i\nu} X_{\nu j} - \sum_{\nu=1}^{m} X_{i\nu} B_{\nu j} = -C_{ij},$$

which means that the entries of the Sylvester operator $\mathcal{S}^{A,B} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$, $X \mapsto AX - XB$ are

$$\mathcal{S}^{A,B}_{ij,pq} = \delta_{jq} A_{ip} - \delta_{ip} B_{qj}, \qquad \delta_{jq} = \begin{cases} 1 & \text{if } j = q, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

If we order the indices columnwise (rowwise), the matrix representation is

$$\mathcal{S}^{A,B}_{col} = \begin{bmatrix} A & & \\ & \ddots & \\ & & A \end{bmatrix} - \begin{bmatrix} B_{11}I & \cdots & B_{m1}I \\ \vdots & \ddots & \vdots \\ B_{1m}I & \cdots & B_{mm}I \end{bmatrix},$$

$$\mathcal{S}^{A,B}_{row} = \begin{bmatrix} A_{11}I & \cdots & A_{1n}I \\ \vdots & \ddots & \vdots \\ A_{n1}I & \cdots & A_{nn}I \end{bmatrix} - \begin{bmatrix} B & & \\ & \ddots & \\ & & B \end{bmatrix}.$$

The Kronecker product

$$X \otimes Y := \begin{bmatrix} X_{11}Y & \cdots & X_{1n}Y \\ \vdots & \ddots & \vdots \\ X_{n1}Y & \cdots & X_{nn}Y \end{bmatrix}$$

allows us to use the short notation

$$\mathcal{S}^{A,B} := \mathcal{S}^{A,B}_{col} = I \otimes A - B^T \otimes I.$$

For the finite element discretization it was advantageous to consider the generalized Sylvester operator

$$\mathcal{S}^{A,B,E} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}, \quad X \mapsto AXE - EXB,$$

which can be written in terms of the Kronecker product by

$$\mathcal{S}^{A,B,E} = E \otimes A - B^T \otimes E;$$

i.e., the entries of the matrix $\mathcal{S}^{A,B,E}$ are

$$\mathcal{S}^{A,B,E}_{ij,pq} = E_{jq}A_{ip} - E_{ip}B_{qj}.$$

**3.2. Analytic structure.** In this section we want to identify the matrices $\mathcal{S}^{A,B}$ and $\mathcal{S}^{A,B,E}$ of the (generalized) Sylvester operator as the discretization of a tensor product operator on the tensor domain $\Omega \times \Omega$. This will enable us to use proofs of multigrid convergence for the product operator.

**3.2.1. Finite element discretization.** We consider the finite element Galerkin discretization of the operator $\mathcal{A} : H^1_0(\Omega \times \Omega) \times H^1_0(\Omega \times \Omega) \to H^{-1}(\Omega \times \Omega)$

$$\mathcal{A}[u](x,y) = -\sum_{\nu=1}^{2} \partial^2_{x_\nu} u(x,y) - \sum_{\nu=1}^{2} \partial^2_{y_\nu} u(x,y) \tag{3.2}$$

using the set $V_{n^2} := \{\varphi_{ij} \mid i,j = 1,\dots,n\}$ of tensor product basis functions based on the basis functions $\phi_i$ from section 1.1.3:

$$\varphi_{ij}(x,y) := \phi_i(x)\phi_j(y), \qquad x,y \in \Omega.$$

The Galerkin stiffness matrix is the matrix $\mathbb{A}$ with entries

$$\begin{aligned}\mathbb{A}_{ij,pq} &= \int_\Omega \int_\Omega \langle \nabla_x \varphi_{ij}(x,y), \nabla_x \varphi_{pq}(x,y)\rangle + \langle \nabla_y \varphi_{ij}(x,y), \nabla_y \varphi_{pq}(x,y)\rangle \,\mathrm{d}x\mathrm{d}y \\ &= \int_\Omega \int_\Omega \langle \nabla \phi_i(x), \nabla \phi_p(x)\rangle \phi_j(y)\phi_q(y) + \langle \nabla \phi_j(y), \nabla \phi_q(y)\rangle \phi_i(x)\phi_p(x) \,\mathrm{d}x\mathrm{d}y \\ &= \int_\Omega \langle \nabla \phi_i(x), \nabla \phi_p(x)\rangle \,\mathrm{d}x \int_\Omega \phi_j(y)\phi_q(y)\,\mathrm{d}y \\ &\quad + \int_\Omega \langle \nabla \phi_j(y), \nabla \phi_q(y)\rangle \,\mathrm{d}y \int_\Omega \phi_i(x)\phi_p(x)\,\mathrm{d}x \\ &= A_{ip}E_{jq} + E_{ip}A_{jq} = \mathcal{S}^{A,A,E}_{ij,pq},\end{aligned}$$

where $A, E$ are the stiffness and mass matrices from section 1.1.3 and $\mathcal{S}^{A,A,E}$ is the Sylvester operator from section 3.1. Therefore, $\mathbb{A}$ is just another notation for $\mathcal{S}^{A,A,E}$, but it allows us to regard it as a standard finite element discretization of an elliptic operator; hence, standard multigrid theory can be applied.

**3.2.2. Finite difference discretization.** For a finite difference discretization of the operator (3.2) one can derive as in the previous section

$$\mathbb{A}^{FD}_{ij,pq} = \mathcal{S}^{A,A}_{ij,pq}, \text{ i.e., } \quad \mathbb{A}^{FD} = \mathcal{S}^{A,A},$$

where $A$ is the finite difference matrix from section 1.1.2.

Before we introduce the multigrid method, we first consider one important ingredient, namely, the smoother. The standard smoother used in a multigrid method is Jacobi (or Gauss–Seidel), which requires in our setting the solution of diagonal Sylvester equations (resp., diagonal generalized Sylvester equations).

## 4. Diagonal Sylvester equation.

A diagonal Sylvester equation

$$\begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix} X - X \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_m \end{bmatrix} + C = 0 \tag{4.1}$$

with $a_i < b_j$ for all $1 \le i \le n$ and $1 \le j \le m$ allows for a direct solution by

$$X_{ij} = C_{ij}/(b_j - a_i). \tag{4.2}$$

If the matrix $C$ is of rank 1 with $R(1)$-matrix representation $C = cd^T$, then

$$X = \begin{bmatrix} c_1 & & \\ & \ddots & \\ & & c_n \end{bmatrix} \begin{bmatrix} (b_1 - a_1)^{-1} & \cdots & (b_m - a_1)^{-1} \\ \vdots & \ddots & \vdots \\ (b_1 - a_n)^{-1} & \cdots & (b_m - a_n)^{-1} \end{bmatrix} \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_m \end{bmatrix}.$$

The following Lemma 4.2 proves that the Cauchy matrix $\mathcal{C}_{ij} = (b_j - a_i)^{-1}$ allows for a low rank approximation (a special case of Theorem 1.3). The idea is to construct a separable representation for the function

$$f(x,y) := \frac{1}{x-y} \approx \sum_{\nu=1}^{k} g_\nu(x) h_\nu(y)$$

so that

$$\mathcal{C}_{ij} \approx \sum_{\nu=1}^{k} g_\nu(b_j)h_\nu(a_i).$$

However, if the distance between the sets

$$I_a := \{a_1, \dots, a_n\} \quad \text{and} \quad I_b := \{b_1, \dots, b_m\}$$

is small compared to their diameters, then the separable approximation requires a large rank $k$. Therefore we subdivide the sets $I_a$ and $I_b$ into subsets $t \subset I_a$ and $s \subset I_b$ so that they fulfil the admissibility condition

$$\min\{\text{diam}(t), \text{diam}(s)\} \le \text{dist}(t, s). \tag{4.3}$$

An explicit construction is given in the following.

CONSTRUCTION 4.1 (local $R(k)$-matrix approximation of the Cauchy matrix). $t \subset I_a$ $s \subset I_b$ (4.3)

$$t_0 := \frac{1}{2}(\min_{a_i \in t} a_i + \max_{a_i \in t} a_i), \quad s_0 := \frac{1}{2}(\min_{b_j \in s} b_j + \max_{b_j \in s} b_j).$$

$i \in t$ $j \in s$

$$\tilde{\mathcal{C}}_{ij} := \begin{cases} \sum_{\nu=0}^{k}(t_0 - b_j)^{-\nu-1}(t_0 - a_i)^\nu & \text{diam}(t) \le \text{diam}(s), \\ \sum_{\nu=0}^{k}(a_i - s_0)^{-\nu-1}(b_j - s_0)^\nu & \end{cases}$$

$\tilde{\mathcal{C}}_{i\in t, j\in s}$ $k$ $U, V$

$$U_{i\nu} := \begin{cases} (t_0 - a_i)^\nu & \text{diam}(t) \le \text{diam}(s), \\ (a_i - s_0)^{-\nu-1} & \end{cases}$$

$$V_{j\nu} := \begin{cases} (t_0 - b_j)^{-\nu-1} & \text{diam}(t) \le \text{diam}(s), \\ (b_j - s_0)^\nu & \end{cases}$$

LEMMA 4.2 (local approximation error). $t$ $s$ $\tilde{\mathcal{C}}$ 4.1 $0 < \varepsilon < 1$

$$|\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| \le \varepsilon |\mathcal{C}_{ij}| \tag{4.4}$$

$i \in t$ $j \in s$

$$k := \lceil \log_3(1/\varepsilon) \rceil + 1.$$

Without loss of generality we assume $\text{diam}(t) \le \text{diam}(s)$. The exact Taylor expansion of $f$ with respect to $x$ is

$$f(x, y) = \sum_{\nu=0}^{\infty} \frac{1}{\nu!} \partial_x^\nu f(t_0, b_j)(a_i - t_0)^\nu = \sum_{\nu=0}^{\infty}(t_0 - b_j)^{-\nu-1}(t_0 - a_i)^\nu.$$

Using this representation and the assumption (4.3) we get

$$\begin{aligned} |\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| &= \Big|\sum_{\nu=k}^{\infty}(t_0 - b_j)^{-\nu-1}(t_0 - a_i)^\nu\Big| \le \sum_{\nu=k}^{\infty} |t_0 - b_j|^{-\nu-1}|t_0 - a_i|^\nu \\ &\le \sum_{\nu=k}^{\infty} \left(\text{dist}(t, s) + \frac{1}{2}\text{diam}(t)\right)^{-\nu} \left(\frac{1}{2}\text{diam}(t)\right)^\nu |t_0 - b_j|^{-1} \\ &\overset{(4.3)}{\le} \sum_{\nu=k}^{\infty} 3^{-\nu}|t_0 - b_j|^{-1} = 3^{-k+1}\frac{1}{2}|t_0 - b_j|^{-1} \le \varepsilon |\mathcal{C}_{ij}|. \qquad \square \end{aligned}$$

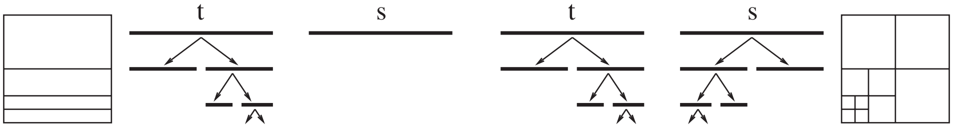


FIG. 4.1. *Recursive subdivision of one (left) or two (right) subintervals and the corresponding partitions of the Cauchy matrix $\mathcal{C}$.*

In order to satisfy the admissibility condition (4.3) there are two strategies.

First, we can subdivide the set $t$ recursively into two parts $t_1$ and $t_2$ of half the diameter so that one of the two is admissible to $s$ (cf. Figure 4.1). The other one is then further subdivided until the diameter is less than the distance to $s$. This strategy produces blocks $t' \times s$, $t' \subset t$, for which we can apply Construction 4.1. The number of blocks is $p := \lceil \log_2(\frac{\mathrm{diam}(t)}{\mathrm{dist}(t,s)}) \rceil + 1$. In total we have to store and compute $\mathcal{O}(pk(n+m))$ entries of the R($k$)-matrix representation.

Second, we can subdivide always both sets $s$ and $t$ each into two parts of half the diameter so that three of the four pairs are admissible and the fourth one has to be subdivided further (cf. Figure 4.1). This strategy will then produce $p := 3\lceil \log_2(\frac{\mathrm{diam}(t)}{\mathrm{dist}(t,s)}) \rceil + 1$ blocks (more than the first strategy), but they are of different size which is decaying geometrically. Therefore we have to store and compute only $\mathcal{O}(k(n+m))$ entries of the R($k$)-matrix representations.

In both cases, the rank of the approximation $\tilde{\mathcal{C}}$ is $pk$. In the second case we can exploit the hierarchical structure for the efficient computation of an approximation for $X$. We will give the details later in Construction 4.5.

COROLLARY 4.3 (approximation error). [illegible] $\tilde{\mathcal{C}}$ [illegible] $\varepsilon$ [illegible] $|\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| \le \varepsilon |\mathcal{C}_{ij}|$ [illegible] $1 \le i \le n$ [illegible] $1 \le j \le m$ [illegible] $C$ [illegible] $R(k_C)$ [illegible] $C_{ij} = \sum_{\nu=1}^{k_C} c_i^{(\nu)} d_j^{(\nu)}$ [illegible] $\tilde{X}_{ij} := \sum_{\nu=1}^{k_C} c_i^{(\nu)} \tilde{\mathcal{C}}_{ij} d_j^{(\nu)}$ [illegible] $X$ [illegible] (4.1) [illegible]

$$|X_{ij} - \tilde{X}_{ij}| \le \varepsilon |X_{ij}|, \qquad \|X - \tilde{X}\|_F \le \varepsilon \|X\|_F.$$

[illegible]

$$|X_{ij} - \tilde{X}_{ij}| = \left| \sum_{\nu=1}^{k_C} c_i^{(\nu)} d_j^{(\nu)} \right| |\tilde{\mathcal{C}}_{ij} - \mathcal{C}_{ij}| \le \varepsilon \left| \sum_{\nu=1}^{k_C} c_i^{(\nu)} d_j^{(\nu)} \right| |\mathcal{C}_{ij}| = \varepsilon |X_{ij}|. \qquad \square$$

REMARK 4.4 (adaptive choice of the rank). In practice, one is interested in a good approximation $\tilde{\mathcal{C}}$ to the Cauchy matrix $\mathcal{C}$, preferably an approximation with minimal rank for a prescribed accuracy $\varepsilon$. Our construction yields only a suboptimal candidate where the rank is higher than necessary. In the multigrid method the approximation $\tilde{\mathcal{C}}$ will be used several times, such that it pays to spend more effort in the computation of $\tilde{\mathcal{C}}$. One way to do this is to compute a candidate $\tilde{\mathcal{C}}_1$ as above up to accuracy $\varepsilon/10$ and compute an approximant $\tilde{\mathcal{C}}$ to $\tilde{\mathcal{C}}_1$ up to accuracy $\varepsilon$ with minimal rank by use of the reduced singular value decomposition of Lemma 2.1.

The construction of a good [illegible] approximant $\tilde{\mathcal{C}}$ to the Cauchy matrix bears two bottlenecks:

First, one has to store a matrix of rank $pk$. For a large scale problem with $n = m = 10^6$, $|b_1 - a_n| = 10^{-3}$, $|a_n - a_1| = 1$, and $\varepsilon = 10^{-6}$, there are more than 300 million entries to be stored, which requires more than two Gigabytes of memory in double precision arithmetic.

Second, the estimated rank *pk* (in the above example $pk = 154$) is typically too large. The truncation to lower rank is of quadratic complexity in the rank, which means prohibitively expensive (cf. Example 2.3).

CONSTRUCTION 4.5 (hierarchical construction). [illegible] $\mathcal{C}$ [illegible] 4.1 [illegible] 1 [illegible] $\mathcal{C}$ [illegible] 2 [illegible] $X$ [illegible] 3 [illegible] $\tilde{X}$ [illegible] $X$

[illegible] 1 [illegible] $t \times s \subset I_a \times I_b$ [illegible] $\tilde{\mathcal{C}}|_{t\times s}$ [illegible] 4.1 [illegible] $k' := \lceil \log_3(\varepsilon^{-1}/3)\rceil + 1$

[illegible] 2 [illegible] $C = \sum_{\nu=1}^{k_C} c^{(\nu)}(d^{(\nu)})^T$ [illegible] $X$ [illegible]

$$X|_{t\times s} = \sum_{\nu=1}^{k_C} \operatorname{diag}(c^{(\nu)}|_t)\, \mathcal{C}|_{t\times s} \operatorname{diag}(d^{(\nu)}|_s),$$

[illegible]

$$\tilde{X}'|_{t\times s} := \sum_{\nu=1}^{k_C} \operatorname{diag}(c^{(\nu)}|_t)\, \tilde{\mathcal{C}}|_{t\times s} \operatorname{diag}(d^{(\nu)}|_s).$$

[illegible] $\tilde{X}''|_{t\times s}$ [illegible]

$$\|\tilde{X}'|_{t\times s} - \tilde{X}''|_{t\times s}\|_F \le \frac{\varepsilon}{3}\|\tilde{X}'|_{t\times s}\|_F.$$

[illegible] 3 [illegible] $\tilde{X}$ [illegible] $I_a \times I_b$ [illegible] $\ell = 1$ [illegible] $\ell = 2$ [illegible] $t \times s$ [illegible]

$$\varepsilon_\ell := 2^{-\ell}\frac{\varepsilon}{3}\|\tilde{X}''\|_F.$$

[illegible] $t \times s$ [illegible] $t_1 \times s_1, t_2 \times s_1, t_1 \times s_2, t_2 \times s_2$ [illegible]

$$\tilde{X}^{t,s} := \mathcal{T}_k\left(\left[\begin{array}{c|c} \tilde{X}''|_{t_1\times s_1} & \tilde{X}''|_{t_1\times s_2} \\ \hline \tilde{X}^{t_2,s_1} & \tilde{X}''|_{t_2\times s_2} \end{array}\right]\right)$$

[illegible] (2.1) [illegible] $\varepsilon_\ell$ [illegible]

[illegible] $\|X - \tilde{X}\|_F / \|X\|_F$ [illegible] $\tilde{X} := \tilde{X}^{I_a, I_b}$ [illegible]

LEMMA 4.6. [illegible] $\tilde{X}$ [illegible] 4.5 [illegible]

$$\|X - \tilde{X}\|_F \le \varepsilon\|X\|_F + \mathcal{O}(\varepsilon^2).$$

[illegible] $\mathcal{O}((n+m)(k_C^2(k')^2 + k_{final}^2))$ [illegible] $k'$ [illegible] $k_C$ [illegible] $C$ [illegible] $k_{final}$ [illegible] $X$

(1) Approximation error. The Cauchy matrix approximation in part 1 of Construction 4.5 was chosen such that $|\mathcal{C}_{ij} - \tilde{\mathcal{C}}_{ij}| \leq \varepsilon|\mathcal{C}_{ij}|/3$. From Corollary 4.3 we conclude $\|\tilde{X}' - X\|_F \leq \varepsilon\|X\|_F/3$. In part 2 of Construction 4.5 the matrix $\tilde{X}'$ is recompressed so that

$$\|X - \tilde{X}''\|_F \leq \|X - \tilde{X}'\|_F + \|\tilde{X}' - \tilde{X}''\|_F \leq \varepsilon\|X\|_F/3 + \varepsilon\|X\|_F/3 + \mathcal{O}(\varepsilon^2).$$

Next we will show that $\|\tilde{X}'' - \tilde{X}\|_F \leq \varepsilon\|X\|_F/3 + \mathcal{O}(\varepsilon^2)$, which gives the desired estimate.

The truncation accuracy in part 3 of Construction 4.5 yields on each level $\ell$ of a block $t \times s$

$$\|\tilde{X}^{t,s} - \tilde{X}''|_{t\times s}\|_F \leq \varepsilon_\ell = 2^{-\ell}\varepsilon\|\tilde{X}''\|_F/3.$$

Over all levels $\ell = 1, \dots$ this sums up to

$$\sum_{\ell=1}^{\infty} \varepsilon_\ell = \frac{1}{3}\varepsilon\|\tilde{X}''\|_F = \frac{1}{3}\varepsilon\|X\|_F + \mathcal{O}(\varepsilon^2).$$

(2) Complexity. Part 1 of Construction 4.5 is of complexity $2^{1-\ell}k'n$ for a block on level $\ell$. On each level there are at most three blocks so that this sums up to

$$\sum_{\ell=1}^{\infty} 2^{1-\ell}3k'n \leq 6k'n.$$

In part 2 we truncate each of the blocks (we neglect the diagonal scaling). Due to Lemma 2.1 the complexity is bounded by

$$\sum_{\ell=1}^{\infty} 2^{1-\ell}3n(k_C k')^2 \leq 6k_C^2(k')^2 n.$$

At last we combine the blocks levelwise. On each level we add four matrices, each of rank at most $k_{final}$, so that the complexity is bounded by

$$\sum_{\ell=1}^{\infty} 2^{1-\ell}nk_{final}^2 \leq 2k_{final}^2 n. \qquad \square$$

In order to illustrate the benefits and the complexity of Construction 4.1 and the alternatives from Remark 4.4 and Construction 4.5, we test the method for a simple artificial model problem.

4.7. The entries of the diagonal matrices $A$ and $B$ are

$$a_i = -i, \qquad b_j = j, \qquad 1 \leq i, j \leq 1024^2.$$

We want to approximate the solution $X$ and the Cauchy matrix $\mathcal{C}$ for a matrix $C$ of rank $k_C = 5$ up to an accuracy of $\varepsilon := 10^{-6}$ by approximations $\tilde{\mathcal{C}}$ and $\tilde{X}$ of minimal rank.

According to Construction 4.5 we compute the approximant in three steps:

1. (Part 1) Hierarchical approximation of $\mathcal{C}$ by $\tilde{\mathcal{C}}$. Since the entries of $\tilde{\mathcal{C}}$ are derived analytically, this is very fast. The blockwise rank $k$ is 15 (as defined in Construction 4.5).

2. (Part 2) Blockwise approximation in the $R(k)$-matrix format.
3. (Part 3) Hierarchical conversion to the $R(k)$-matrix format.

The following table displays the times the three steps take and the amount of storage needed (in the first step for $\tilde{\mathcal{C}}$ and in the second and third step for $\tilde{X}''$ and $\tilde{X}$, respectively). The numerical tests were performed on a SUN UltraSPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate.

| | time (seconds) | storage (Megabyte) |
|---|---|---|
| Part 1 | 10.3 | 720 |
| Part 2 | 943 | 180 |
| Part 3 | 422 | 360 |

The amount of storage needed in step 1 can be omitted by immediate truncation of each block to lower rank. The final approximation $\tilde{X}$ has a rank of $k_{\tilde{X}} = 22$.

The previous example illustrates that the hierarchical truncation is an efficient way to generate a best approximation either to the Cauchy matrix or to the solution of a diagonal Sylvester equation. In practice, we will use the construction to solve diagonal Sylvester equations as they appear in the Jacobi iteration. There the diagonal entries are of similar size; i.e., the matrix is well-conditioned such that the number of levels is small (typically one). The situation simplifies if all diagonal entries are equal.

4.8. The entries of the diagonal matrices $A$ and $B$ are

$$a_i \equiv a, \qquad b_j \equiv b, \qquad 1 \le i \le n, 1 \le j \le m.$$

Then the Cauchy matrix is $\mathcal{C} = (b-a)^{-1} I$ ($I$ is the identity), and the solution is $X = (b-a)^{-1} C$.

In the following example we want to compare our construction with an iterative scheme that approximates the solution $X$ to (4.1). For this example we fix a matrix $C$ of rank $k_C := 5$.

4.9. The entries of the diagonal matrices $A$ and $B$ are

$$a_i = -i, \qquad b_j = j, \qquad 1 \le i, j \le 1024^2.$$

The ADI iteration from [18] to solve $AX - XB + C = 0$ starts with $X_0 := 0$ and generates the matrices

$$\begin{aligned} X_{i+1} := {} & (A - p_i I)(A + p_i I)^{-1}\, X_i\, (A - p_i I)(A + p_i I)^{-1} \\ & -2p_i (A + p_i I)^{-1}\, C\, (A + p_i I)^{-1}, \end{aligned}$$

where the parameters $p_i$ for $J$ steps of the iteration are given by $\kappa := (a_1/a_n)^{1/J}, t_0 := a_1, t_j := \kappa\, t_{j-1}, p_j := -\sqrt{t_{j-1} t_j}$ for $j = 1, \ldots, J$. This parameter choice allows for an explicit bound on the relative error, such that the number $J$ of steps can be determined a priori. The rank of the resulting approximant $\tilde{X}^{ADI}$ to $X$ is equal to $Jk_C$. In this example the number of iterations $J = 81$ ensures that the a priori error bound is less than $10^{-6}$.

The computation of an approximation $\tilde{X}^{ADI}$ takes ca. 580 seconds (this time can be reduced by using the ADI variant from [15]). However, the rank used in the representation of $\tilde{X}^{ADI}$ is $k = 405$ so that a truncation to smaller (minimal) rank would require approximately 25000 seconds (cf. Table 2.1). Alternatively, one could truncate in intermediate steps (no control of the accuracy) so that the time reduces to approximately 5000 seconds. The complexity is higher than for the hierarchical Construction 4.5 because the local blockwise ranks are much smaller than the global rank $Jk_C$ from the ADI iteration.

**4.1. Diagonal generalized Sylvester equation.** At last we want to comment on diagonal [illegible] Sylvester equations. There the system

$$\begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix} X \begin{bmatrix} e_1 & & \\ & \ddots & \\ & & e_m \end{bmatrix} - \begin{bmatrix} \hat{e}_1 & & \\ & \ddots & \\ & & \hat{e}_n \end{bmatrix} X \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_m \end{bmatrix} + C = 0$$

has to be solved. We assume that $e_j > 0, \hat{e}_i > 0, a_i < 0$, and $b_j > 0$ for all entries of the diagonal matrices. This system can, by multiplication with $\mathrm{diag}(\hat{e}_1^{-1}, \dots, \hat{e}_n^{-1})$ from the left and $\mathrm{diag}(e_1^{-1}, \dots, e_m^{-1})$ from the right, be transformed into a standard Sylvester equation for which the techniques from above are applicable, in particular

$$X_{ij} = \hat{e}_i^{-1} C_{ij} e_j^{-1} / (b_j/e_j - a_i/\hat{e}_i). \tag{4.5}$$

**5. Smoothing iterations.** In this section we will consider possible smoothing iterations that are useful in the context of the multigrid method. The two simplest ones are Richardson and Jacobi, and these will be given in detail in the following.

**5.1. Richardson iteration.** For linear systems of equations $Mx = b$ the (stationary) Richardson iteration is defined by

$$x_0 := 0, \qquad x_i := x_{i-1} - \theta(Mx_{i-1} - b) \qquad \text{for } i \geq 1.$$

Convergence is guaranteed for positive definite matrices $M$ if the parameter $\theta \in \mathbb{R}$ fulfils $0 < \theta < 2\|M\|_2^{-1}$ (see, e.g., [8] and also for a generalization to nonsymmetric systems). For the linear system $AX - XB + C = 0$ the iteration reads

$$X_0 := 0, \qquad X_i := X_{i-1} - \theta(AX_{i-1} - X_{i-1}B + C) \qquad \text{for } i \geq 1. \tag{5.1}$$

The optimal damping factor is $\theta = 2/(\|M\|_2 + \|M^{-1}\|_2^{-1})$, which can be estimated by $\theta \approx \frac{3}{2}\|M\|_2^{-1}$, where $\|M\|_2 = \|A\|_2 + \|B\|_2$ is easily computable via the power iteration (here we assumed $A < 0$ and $B > 0$).

If the iterate $X_{i-1}$ is an $R(k)$-matrix and the right-hand side $C$ is an $R(k_C)$-matrix, then the next iterate $X_i$ is an $R(2k + k_C)$-matrix whose representation can be computed by $k$ matrix-vector multiplications for the matrices $A$ and $B$. In order to stay in the set of $R(k)$-matrices one can truncate the resulting matrix $X_i$ to lower rank $k$. This will be called the $R(k)$-Richardson iteration:

$$X_0 := 0, \qquad X_i := \mathcal{T}_k(X_{i-1} - \theta(AX_{i-1} - X_{i-1}B + C)) \qquad \text{for } i \geq 1. \tag{5.2}$$

Lemma 5.1. [illegible] $A \in \mathbb{R}^{n \times n}$ [illegible] $B \in \mathbb{R}^{m \times m}$ [illegible] $A$ [illegible] $B$ [illegible] $\mathcal{O}(n)$ [illegible] $\mathcal{O}(m)$

(a) [illegible] (5.1) [illegible] $\mathcal{O}(nm)$

(b) [illegible] $R(k)$ [illegible] (5.2) [illegible] $\mathcal{O}(k^2(n+m))$

Although the Richardson iteration is convergent for sufficiently small $\theta$, the rate of convergence can be poor. In the context of multigrid methods one is not necessarily interested in convergence properties but in the smoothing property (cf. [8]). The next lemma provides the necessary assumptions.

Lemma 5.2. [illegible] $A \in \mathbb{R}^{n \times n}$ [illegible] $B \in \mathbb{R}^{m \times m}$ [illegible] $\sigma(A) > \sigma(B)$ [illegible] $E$ [illegible]

[illegible] $\mathcal{S}^{A,B}$ [illegible] $\mathcal{S}^{A,B,E}$ [illegible]

[illegible] If $\sigma(A) = \{\lambda_1, \dots, \lambda_n\}$ and $\sigma(B) = \{\mu_1, \dots, \mu_m\}$, then the $nm$ eigenvalues of the (linear) Sylvester operator $S : X \mapsto AX - XB$ are $\lambda_i - \mu_j$. By assumption all eigenvalues are positive. The symmetry follows from $\mathcal{S}^{A,B}_{ij,pq} = \delta_{jq}A_{ip} - \delta_{ip}B_{qj} = \delta_{qj}A_{pi} - \delta_{pi}B_{jq} = \mathcal{S}^{A,B}_{pq,ij}$. Analogously symmetry holds for the generalized Sylvester operator $\mathcal{S}^{A,B,E}$. From $\sigma(A) > \sigma(B)$ and the symmetry of $A, B$ we conclude $\sigma(E^{-\frac{1}{2}}AE^{-\frac{1}{2}}) > \sigma(E^{-\frac{1}{2}}BE^{-\frac{1}{2}})$. From the first part we know $S = I \otimes E^{-\frac{1}{2}}AE^{-\frac{1}{2}} - E^{-\frac{1}{2}}BE^{-\frac{1}{2}} \otimes I > 0$ and thus by multiplying $E^{\frac{1}{2}} \otimes E^{\frac{1}{2}}$ from the left and right: $\mathcal{S}^{A,B,E} = E \otimes A - B \otimes E > 0$. ☐

**5.2. Jacobi iteration.** The [illegible] is defined by

$$X_0 := 0, \qquad X_i := X_{i-1} - \theta \operatorname{diag}(\mathcal{S})^{-1}(AX_{i-1} - X_{i-1}B + C) \qquad \text{for } i \geq 1, \tag{5.3}$$

where $\mathcal{S}$ is the Sylvester operator. The diagonal entries of the Sylvester operator are

$$\mathcal{S}^{A,B}_{ij,ij} = A_{ii} - B_{jj}, \quad \mathcal{S}^{A,B,E}_{ij,ij} = E_{jj}A_{ii} - E_{ii}B_{jj},$$

so that the corresponding Sylvester equations are

$$\operatorname{diag}(A)X - X\operatorname{diag}(B) = C_i, \quad \operatorname{diag}(A)X\operatorname{diag}(E) - \operatorname{diag}(E)X\operatorname{diag}(B) = \tilde{C}_i.$$

We have to solve the diagonal (generalized) Sylvester equations for the right-hand side $C_i = AX_{i-1} - X_{i-1}B + C$ and $\tilde{C}_i = AX_{i-1}E - EX_{i-1}B + C$, respectively. The solution is given by (4.2) and (4.5). The optimal damping factor for the Jacobi iteration is $\theta := 2/(\Lambda + \lambda)$ [8], where $\Lambda$ and $\lambda$ are the best bounds for

$$\lambda \operatorname{diag}(M) \leq M \leq \Lambda \operatorname{diag}(M), \qquad M = \mathcal{S}^{A,B} \text{ or } \mathcal{S}^{A,B,E}.$$

Later we will use the parameter $\theta := 1/2$, which is sufficient to guarantee the smoothing property [8] needed for the multigrid method.

If the iterate $X_{i-1}$ is an $R(k)$-matrix and the right-hand side $C$ is an $R(k_C)$-matrix, then the right-hand side is an $R(2k + k_C)$-matrix. A low rank approximation $X_{i+1}$ to the solution of the diagonal Sylvester equation can be computed by means of the hierarchical Construction 4.5. The effect is the same if we solve the diagonal equation exactly and truncate the result to a fixed rank $k$ or a fixed accuracy $\varepsilon$. Therefore, the $R(k)$-Jacobi iteration can be written in the form

$$X_0 := 0, \qquad X_i := \mathcal{T}_k(X_{i-1} - \theta \operatorname{diag}(\mathcal{S})^{-1}(AX_{i-1} - X_{i-1}B + C)). \tag{5.4}$$

LEMMA 5.3. [illegible] $A \in \mathbb{R}^{n \times n}$ [illegible] $B \in \mathbb{R}^{m \times m}$ [illegible] $A$ [illegible] $B$ [illegible] $\mathcal{O}(n)$ [illegible] $\mathcal{O}(m)$

(a) [illegible] (5.3) [illegible] $\mathcal{O}(nm)$

(b) [illegible] $R(k)$ [illegible] (5.4) [illegible] $\mathcal{O}(k^2(n+m))$

**5.3. ADI iteration as a smoother.** Apart from Richardson and Jacobi there are many other popular smoothers such as Gauss–Seidel, SOR, ILU, etc. Since these are not compatible with the low rank format, they are not of interest here. The only notable exception that we are aware of is the ADI iteration from Example 4.9. There we have to solve systems of the form

$$Ax = b,$$

which can be accomplished, e.g., by a multigrid method. However, one has to be careful with the choice of the shift parameters $p_i$, since the optimal parameters for the smoothing property differ from the usual ones that yield the optimal convergence rate [7].

For sure, the Richardson iteration is the most simple of the smoothers under consideration. The Jacobi iteration is necessary for nonuniform grids (e.g., locally refined) in order to get mesh-independent good convergence rates. The same goal is reached by the ADI iteration.

**6. Multigrid method.** The Richardson and Jacobi iterations introduced in the previous section smooth the defect in the multigrid method on one level (=grid). In the multigrid method we transfer the smoothed defect to a coarser grid and compute a defect correction on the coarser grid. The coarse grid correction is then transferred to the fine grid in order to reduce the smooth parts of the defect. On the coarsest level we use a standard solution for the Sylvester equation. For the transfer between different grids ranging from coarse ($n_0 = 9$ degrees of freedom) to fine ($n_8 = 1046529$ degrees of freedom), we need the prolongation and restriction operator defined in the following. Whereas the Richardson and Jacobi iteration had to be adopted to the low rank setting, this is not necessary for the grid transfer operators.

Let $X \in \mathbb{R}^{n\times n}$ be a matrix, and let $\hat{p} : \mathbb{R}^n \to \mathbb{R}^m$ be a linear mapping, the so-called prolongation. Then the corresponding matrix mapping $p$ is defined by

$$p(X) := \hat{p}X\hat{p}^T. \tag{6.1}$$

The adjoint operator $r$, the so-called restriction, is given by

$$r(Y) := \hat{p}^T Y \hat{p} \tag{6.2}$$

for $Y \in \mathbb{R}^{m\times m}$. Since the linear mapping $p$ does not increase the rank of a matrix, we stay in the set of $R(k)$ matrices (only of different size $n, m$). Moreover, if $X = AB^T$ is an $R(k)$-matrix, then

$$p(X) = (\hat{p}A)(\hat{p}B)^T,$$

so that the prolonged (or restricted in the case $r(Y)$) matrix is naturally given in the desired $R(k)$ format. In the notation of section 3.1 the prolongation is of the tensor structure $p = \hat{p} \otimes \hat{p}$.

**6.1. Multigrid algorithm and convergence results.** Let $\ell = 0, \dots, L$ be the level numbers, and assume that on each level we have a discrete linear equation[1]

$$\mathfrak{A}_\ell \mathfrak{x}_\ell = \mathfrak{b}_\ell \qquad (0 \le \ell \le L),$$

with symmetric and positive definite $n_\ell \times n_\ell$ matrices $\mathfrak{A}_\ell$, while $\mathfrak{b}_\ell$ is some right-hand side and $\mathfrak{x}_\ell$ the corresponding solution. For some domains (e.g., the unit square from our model problem) the hierarchy $(\mathfrak{A}_\ell)_{\ell=0}^L$ of discrete problems is naturally given by successive refinement of the coarsest grid. For more complicate domains one needs suitable coarsening algorithms, e.g., composite finite elements [10] or algebraic multigrid [19, 21].

We recall the general multigrid algorithm (for details of the algorithm or the following statements we refer to Hackbusch [7], [8]):

---

[1] The Fraktur style letters indicate matrices and vectors which will be later identified with corresponding quantities of the Sylvester equation. For instance, the vector $\mathfrak{x}_\ell$ will become the unknown solution matrix $X_\ell$.

```
function $MGM(\ell,\mathfrak{x},\mathfrak{b})$;                              (returns the new iterate)
if $\ell = 0$ then $\mathfrak{x} := \mathfrak{A}_0^{-1} f$ else
begin
    for $i := 1$ to $\nu$ do $\mathfrak{x} := \mathcal{S}_\ell(\mathfrak{x},\mathfrak{b})$;                         (presmoothing)
    $\mathfrak{d} := r(\mathfrak{A}_\ell \mathfrak{x} - \mathfrak{b})$;                             (restriction of the defect)
    $\mathfrak{y} := 0$;                                  (starting value for the corrections)
    for $i := 1$ to $\gamma$ do $v := MGM(\ell - 1, \mathfrak{y}, \mathfrak{d})$;
    $\mathfrak{x} = \mathfrak{x} - p\mathfrak{y}$;                                       (coarse-grid correction)
    for $i := 1$ to $\nu$ do $\mathfrak{x} := \mathcal{S}_\ell(\mathfrak{x},\mathfrak{b})$;                        (postsmoothing)
end;
$MGM := \mathfrak{x}$                                                 (new iterate returned)
```

The $V$-cycle ($W$-cycle) corresponds to $\gamma = 1$ ($\gamma = 2$). $\nu$ is the number of pre- and postsmoothing steps using the smoothing procedure $\mathcal{S}_\ell$ (e.g., Richardson, Jacobi, or the $R(k)$ counterparts). $p$ is the prolongation from (6.1), e.g., the piecewise linear interpolation in the case of difference schemes, or the canonical finite element transfer in the case of finite element subspaces $V_{\ell-1} \subset V_\ell$).

The essential conditions for the convergence of the $W$-cycle are the smoothing and approximation properties. A simplified version of the smoothing property is

$$\|\mathfrak{A}_\ell \mathfrak{S}_\ell^\nu\| \leq C_{\mathrm{sm}} \|\mathfrak{A}_\ell\| \, \sigma_\ell \eta(\nu) \qquad \text{for } \nu \geq 1 \text{ with } \lim_{\nu\to\infty} \eta(\nu) = 0, \tag{6.3}$$

where $\mathfrak{S}_\ell$ is the iteration matrix of the iteration $\mathcal{S}_\ell$ (i.e., $\mathcal{S}_\ell(\mathfrak{x},\mathfrak{b}) = \mathfrak{S}_\ell\mathfrak{x} + \mathfrak{T}_\ell\mathfrak{b}$) and $C_{\mathrm{sm}}$ is a constant independent of $\ell$, while $\sigma_\ell$ is any scaling quantity (except of section 6.6, only $\sigma_\ell = 1$ will occur). For convenience, $\|\cdot\|$ may be considered as the spectral norm, but other norms are possible. Often $\eta(\nu)$ equals

$$\eta_0(\nu) := \nu^\nu / (\nu+1)^{\nu+1}. \tag{6.4}$$

The approximation property reads

$$\|\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1} r\| \leq C_{\mathrm{app}} / (\|\mathfrak{A}_\ell\| \, \sigma_\ell), \tag{6.5}$$

with an $\ell$-independent constant $C_{\mathrm{app}}$ and the same scaling quantity $\sigma_\ell$ as in (6.3). Under these assumptions (and simple technical conditions on $p$, $r$, and $\mathfrak{S}_\ell$), the $W$-cycle converges with the rate $\mathrm{const} \cdot \eta(\nu)$ (under standard symmetry conditions on $p$, $r$, and $\mathfrak{S}_\ell$, even $\nu = 1$ leads to convergence).

**6.2. Approximation property.** Assuming a finite element discretization with subspaces $V_{\ell-1} \subset V_\ell$ with quasi-uniform grid sizes $h_{\ell-1}$, $h_\ell$ ($h_{\ell-1}/h_\ell \leq \mathrm{const}$), one obtains the estimate $\|\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1} r\| \leq \mathrm{const} \cdot \|\mathfrak{A}_\ell^{-1}\| h_\ell^2$ for the spectral norm, provided that full regularity holds; i.e., the underlying boundary value problem satisfies $\|u\|_{H^2(\omega)} \leq \mathrm{const}\, \|f\|_{L^2(\omega)}$ for the solution of $Lu = f$. If the coefficients are sufficiently smooth and $\omega$ is convex (or an image of a convex domain under a smooth mapping), full regularity holds (cf. Hackbusch [9, Theorem 9.1.22]). In our application, the domain $\omega$ is the product $\Omega \times \Omega$. Convexity of $\Omega$ implies convexity of $\omega$.

Since the scaling of the stiffness matrix is such that $\|\mathfrak{A}_\ell\| \|\mathfrak{A}_\ell^{-1}\|$ is proportional to $h_\ell^{-2}$, the inequality $\|\mathfrak{A}_\ell^{-1} - p\mathfrak{A}_{\ell-1}^{-1} r\| \leq \mathrm{const} \cdot \|\mathfrak{A}_\ell^{-1}\| h_\ell^2$ is equivalent to (6.5) with $\sigma_\ell := 1$.

Weaker regularity can also be treated (see section 6.6).

**6.3. Smoothing property without truncation.** First we consider the Richardson iteration

$$\mathcal{S}_\ell(\mathfrak{x}_\ell, \mathfrak{b}_\ell) = \mathfrak{S}_\ell \mathfrak{x}_\ell + \mathfrak{T}_\ell \mathfrak{b}_\ell, \qquad \text{with } \mathfrak{S}_\ell = I - \vartheta_\ell \mathfrak{A}_\ell, \quad \mathfrak{T}_\ell \mathfrak{b}_\ell = \vartheta_\ell \mathfrak{b}_\ell,$$

with the damping factor $\vartheta_\ell = 1/\|\mathfrak{A}_\ell\|_2$ (also $\vartheta_\ell = 1/\rho(\mathfrak{A}_\ell)$ because of the symmetry of $\mathfrak{A}_\ell$).

Lemma 6.1 (see [8, Theorem 10.6.5]). [illegible] $\mathfrak{A}_\ell$ [illegible] $\vartheta_\ell = 1/\|\mathfrak{A}_\ell\|_2$ [illegible] $\eta(\nu) = \eta_0(\nu)$ [illegible] (6.4) [illegible] $C_{\mathrm{sm}}\sigma_\ell = 1$.

For the application to the Sylvester equation, we have to make use of $\vartheta_\ell = 1/\rho(\mathfrak{A}_\ell) = 1/(\max_{\lambda\in\sigma(A)} \lambda - \min_{\mu\in\sigma(B)} \mu)$. Since the matrices $\mathfrak{A}_\ell = \mathcal{S}^{A,B}$ and $\mathfrak{A}_\ell = \mathcal{S}^{A,B,M}$ are symmetric and positive definite (Lemma 5.2), the previous lemma applies.

Next, we consider the damped Jacobi iteration

$$\mathcal{S}_\ell(\mathfrak{x}_\ell, \mathfrak{b}_\ell) = \mathfrak{S}_\ell \mathfrak{x}_\ell + \mathfrak{T}_\ell \mathfrak{b}_\ell, \qquad \text{with } \mathfrak{S}_\ell = I - \vartheta_\ell \mathfrak{D}_\ell^{-1} \mathfrak{A}_\ell, \quad \mathfrak{T}_\ell \mathfrak{b}_\ell = \vartheta_\ell \mathfrak{D}_\ell^{-1} \mathfrak{b}_\ell, \tag{6.6}$$

where $\mathfrak{D}_\ell$ is the diagonal part of $\mathfrak{A}_\ell$ with $\vartheta_\ell$ such that $\vartheta_\ell \rho(\mathfrak{D}_\ell^{-1}\mathfrak{A}_\ell) \le 1$. Then we obtain the following.

Lemma 6.2 (see [7, section 6.2]). [illegible] $\mathfrak{A}_\ell$ [illegible] (6.6) [illegible]

$$\vartheta_\ell \le 1/\rho(\mathfrak{D}_\ell^{-1}\mathfrak{A}_\ell) \quad [illegible] \quad \|\mathfrak{D}_\ell\|_2 \le C_{\mathrm{sm}} \vartheta_\ell \|\mathfrak{A}_\ell\|_2 \tag{6.7}$$

[illegible] $\eta(\nu) = \eta_0(\nu)$ [illegible] $C_{\mathrm{sm}}\sigma_\ell = 1$. [illegible] $\mathfrak{D}_\ell$ [illegible] (6.7)

The Jacobi iteration (5.3) for the Sylvester equation with suitable $\theta$ satisfies the assumptions of the previous lemma (due to Lemma 5.2); therefore, the smoothing property holds. In the case that the matrix $\mathfrak{D}_\ell$ is replaced by an approximation $\mathfrak{D}'_\ell$ due to the fact that we solve the diagonal Sylvester equation with a perturbed Cauchy matrix, we chose a symmetric $R(k)$-approximation of the Cauchy matrix. Since $\mathfrak{D}_\ell$ is well-conditioned, the approximation remains positive definite and satisfies the inequalities (6.7) with a possibly modified constant $C_{\mathrm{sm}}$. Hence, the Jacobi iteration with $R(k)$-Cauchy matrix approximation also possesses the smoothing property. In combination with the approximation property from above, we obtain level-independent convergence of the $W$-cycle. Similarly, the $V$-cycle proof from [7] can be applied. The iterates $X_i$ are all treated as full matrices, and the multigrid method therefore has a complexity of $\mathcal{O}(n_\ell)$, where $X_i \in \mathbb{R}^{\sqrt{n_\ell}\times\sqrt{n_\ell}}$.

**6.4. Truncation of the iterates.** The effect of the truncation in each step of the multigrid method (during the smoothing iteration and defect correction) can be regarded as an artificially limited machine precision. After one full multigrid cycle, the $i$th iterate $\mathfrak{x}_\ell^i$ on level $\ell$ is perturbed by $\mathfrak{s}_\ell^i$ so that the equation

$$\mathfrak{A}_\ell \mathfrak{x}_\ell^i = \mathfrak{b}_\ell - \mathfrak{A}_\ell \mathfrak{s}_\ell^i$$

holds. The vector $\mathfrak{s}_\ell^i$ accumulates all of the perturbations of size $\varepsilon$ during the $i$th cycle (due to the truncation to a fixed rank), $\|\mathfrak{s}_\ell^i\| \le C_{n_\ell}\varepsilon$. Since we expect a convergence rate of

$$\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\| < \rho \|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\|, \qquad \rho < 1,$$

we immediately get

$$\|(\mathfrak{x}_\ell^i + \mathfrak{s}_\ell^i) - \mathfrak{x}_\ell\| \;<\; \rho\|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\| + C_{n_\ell}\varepsilon \;\le\; \underbrace{\left(\rho + \frac{C_{n_\ell}\varepsilon}{\|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\|}\right)}_{\tilde\rho} \|\mathfrak{x}_\ell^{i-1} - \mathfrak{x}_\ell\|.$$

As long as we can represent $\mathfrak{x}_\ell^{i-1}$ sufficiently well, the term $C_{n_\ell}\varepsilon$ is small and the perturbed convergence rate $\tilde\rho \approx \rho$. As we come closer to the solution $\mathfrak{x}_\ell$ and fix the accuracy $\varepsilon$ (the rank $k$, resp.), the convergence will break down (the iteration stagnates). This is also observed in the numerical tests.

REMARK 6.3 (nonsymmetry). Although the analysis given here requires the symmetry of the system matrix $\mathfrak{A}_\ell$ (which is induced by the symmetry of $A$ and $B$), the multigrid method still works well for the nonsymmetric case.

In principle, the whole machinery of (algebraic) multigrid methods can be transferred to the Sylvester case by the tensor product relation. The crucial point is the connection between the hierarchies of the $A$ matrices and the $B$ matrices since these are generated independently. This "natural" anisotropy is considered in the next section.

**6.5. Anisotropic case.** As seen from (3.2) and the derived approximation, the stiffness matrix $\mathbb{A}$ is the sum $\mathbb{A}_x + \mathbb{A}_y$, where $\mathbb{A}_x, \mathbb{A}_y$ belong to the $x$- and $y$-variables. In the case of (3.2), the differential operator $\mathcal{A}[u](x,y)$ is $\mathcal{A} = -\Delta_x - \Delta_y$; i.e., the $x$- and $y$-parts are equal so that $\mathbb{A}_x$ and $\mathbb{A}_y$ have identical eigenvalues. A typical anisotropic differential operator is $\mathcal{A} = -\Delta_x - \varepsilon\Delta_y$, with small, positive $\varepsilon$. In this case the approximation property contains an additional factor $1/\varepsilon$, which may be formulated by the choice $\sigma_\ell = \varepsilon$. Therefore, we need a smoothing procedure such that the estimate (6.3) is improved by the same factor $\sigma_\ell = \varepsilon$. This can be obtained by the iteration

$$\mathcal{S}_\ell(\mathfrak{x}_\ell, \mathfrak{b}_\ell) = \mathfrak{S}_\ell\mathfrak{x}_\ell + \mathfrak{T}_\ell\mathfrak{b}_\ell, \qquad \text{with } \mathfrak{S}_\ell = I - \mathfrak{A}_{x,\ell}^{-1}\mathfrak{A}_\ell, \quad \mathfrak{T}_\ell\mathfrak{b}_\ell = \mathfrak{A}_{x,\ell}^{-1}\mathfrak{b}_\ell, \tag{6.8}$$

where $\mathfrak{A}_\ell = \mathfrak{A}_{x,\ell} + \varepsilon\mathfrak{A}_{y,\ell}$. The terms $\mathfrak{A}_{x,\ell}$ and $\varepsilon\mathfrak{A}_{y,\ell}$ are the discretizations $\mathbb{A}_x, \mathbb{A}_y$ at level $\ell$.

LEMMA 6.4 (see [7, Lemma 10.1.2]). [illegible] $\mathfrak{A}_{x,\ell}, \mathfrak{A}_{y,\ell}$ [illegible]

$$\mathfrak{A}_{x,\ell} \cdot \mathfrak{A}_{y,\ell} = \mathfrak{A}_{y,\ell} \cdot \mathfrak{A}_{x,\ell}, \qquad \|\mathfrak{A}_{y,\ell}\|_2 \le \text{const}\ \|\mathfrak{A}_\ell\|_2\,.$$

[illegible] (6.8) [illegible] $\eta(\nu) = \eta_0(\nu - 1)$ [illegible] $C_{\mathrm{sm}}\sigma_\ell = \varepsilon$.

[illegible] For convenience, we repeat the proof, which is based on the identity

$$\begin{aligned}
\mathfrak{A}_\ell\mathfrak{S}_\ell^\nu &= (\mathfrak{A}_{x,\ell} + \varepsilon\mathfrak{A}_{y,\ell})\,\mathfrak{S}_\ell^\nu = \mathfrak{A}_{x,\ell}\mathfrak{S}_\ell^\nu + \varepsilon\mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu = \mathfrak{A}_{x,\ell}\left(I - \mathfrak{A}_{x,\ell}^{-1}\mathfrak{A}_\ell\right)\mathfrak{S}_\ell^{\nu-1} + \varepsilon\mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu \\
&= (\mathfrak{A}_{x,\ell} - \mathfrak{A}_\ell)\,\mathfrak{S}_\ell^{\nu-1} + \varepsilon\mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu = -\varepsilon\mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^{\nu-1} + \varepsilon\mathfrak{A}_{y,\ell}\mathfrak{S}_\ell^\nu \\
&= \varepsilon\mathfrak{A}_{y,\ell}\,(I - \mathfrak{S}_\ell)\,\mathfrak{S}_\ell^{\nu-1}.
\end{aligned}$$

The commutativity is used to show that $\mathfrak{S}_\ell$ is symmetric with eigenvalues in $[0,1]$. This implies $\|(I - \mathfrak{S}_\ell)\mathfrak{S}_\ell^{\nu-1}\|_2 = \rho((I - \mathfrak{S}_\ell)\mathfrak{S}_\ell^{\nu-1}) \le \eta_0(\nu - 1)$ (cf. [8, Lemma 10.6.1]). The final result follows from $\|\mathfrak{A}_\ell\mathfrak{S}_\ell^\nu\|_2 = \varepsilon\|\mathfrak{A}_{y,\ell}(I - \mathfrak{S}_\ell)\mathfrak{S}_\ell^{\nu-1}\|_2 \le \varepsilon\|\mathfrak{A}_{y,\ell}\|_2$ $\|(I - \mathfrak{S}_\ell)\mathfrak{S}_\ell^{\nu-1}\|_2 \le \varepsilon \cdot \text{const} \cdot \|\mathfrak{A}_\ell\|_2\eta_0(\nu - 1)$. $\square$

Since the extra factor $\varepsilon$ from the smoothing property cancels with the $1/\varepsilon$ factor in the approximation property, the estimate of the multigrid convergence rate is independent of $\ell$ and $\varepsilon$.

The commutativity $\mathfrak{A}_{x,\ell} \cdot \mathfrak{A}_{y,\ell} = \mathfrak{A}_{y,\ell} \cdot \mathfrak{A}_{x,\ell}$ holds in the case of the Sylvester equation because of the tensor structure.

The solution of the system $\mathfrak{A}_{x,\ell}$ in each step of the iteration (6.8) requires the solution of a system

$$A_\ell X_\ell = C_\ell, \qquad X_\ell \in \mathbb{R}^{\sqrt{n_\ell} \times \sqrt{n_\ell}},$$

which can be done columnwise for full matrices or just for the $k$ column vectors of $U$ in the $R(k)$-matrix representation of Definition 1.2.

**6.6. Weaker regularity.** In the case of a reentrant corner of $\Omega$, the full regularity is not satisfied, but $\mathcal{A} : H^{2-s}(\Omega \times \Omega) \to H^{-s}(\Omega \times \Omega)$ is an isomorphism for some $s \in [0, 1)$ (cf. [9]). In this case, one has to modify the matrix norms in (6.3) and (6.5):

$$\left\| \mathfrak{A}_\ell^{1-s/2} \mathfrak{S}_\ell^\nu \, \mathfrak{A}_\ell^{-s/2} \right\|_2 \le C_{\mathrm{sm}} \left\| \mathfrak{A}_\ell \right\|_2^{1-s} \sigma_\ell \eta(\nu), \tag{6.9a}$$

$$\| \mathfrak{A}_\ell^{-s/2} \left( \mathfrak{A}_\ell^{-1} - p \mathfrak{A}_{\ell-1}^{-1} r \right) \mathfrak{A}_\ell^{s/2} \|_2 \le C_{\mathrm{app}} / \left( \| \mathfrak{A}_\ell \|_2^{1-s} \sigma_\ell \right), \tag{6.9b}$$

involving fractional powers of $\mathfrak{A}_\ell$. In the case of $\mathfrak{A}_{x,\ell} \cdot \mathfrak{A}_{y,\ell} = \mathfrak{A}_{y,\ell} \cdot \mathfrak{A}_{x,\ell}$, it follows that $\mathfrak{A}_\ell^\alpha$ $(0 \le \alpha \le 1)$, with $\mathfrak{A}_\ell = \mathfrak{A}_{x,\ell} + \varepsilon \mathfrak{A}_{y,\ell}$, is spectrally equivalent with $\mathfrak{A}_{x,\ell}^\alpha + \varepsilon^\alpha \mathfrak{A}_{y,\ell}^\alpha$. Using these matrices in (6.9) instead of $\mathfrak{A}_\ell^\alpha$, one can repeat the proof of Lemma 6.4 with $\sigma_\ell = \varepsilon^{1-s}$ and $\eta(\nu) = (\eta_0(\nu))^{1-s}$.

**7. Numerical results.** In this section we apply the $R(k)$-multigrid algorithm developed in the previous sections to the model problem of section 1.1 with $\omega := 1$ in (1.1), namely, the first step of the Newton method where we have to solve (1.6) with a rank one right-hand side $C$ and the two-dimensional discrete Laplacian $A$. The simple geometry allows us to use two-dimensional bilinear (tensor) basis functions $\phi_i$ and a nested hierarchy of grids with a coarse grid that contains $n_0 = 9$ degrees of freedom; see Figure 7.1.

The computations are performed on a SUN ULTRASPARC III with 900 MHz CPU clock rate and 150 MHz memory clock rate. We make use of the LAPACK and BLAS libraries (`http://www.netlib.org`) for the truncation procedure and use the standard C programming language otherwise.

The initial approximation on level $\ell$ is obtained by prolongation of a solution from level $\ell - 1$; i.e., we use a nested iteration so that only $\mathcal{O}(1)$ steps on the fine grid are necessary in order to reduce the error down to the size of the discretization error. The rank on level $\ell - 1, \ldots, 0$ is chosen as twice the rank $k$ on the fine grid $\ell$ on which we want to compute the solution. In the $V$-cycle multigrid we use $\nu = 2$ pre- and postsmoothing steps.

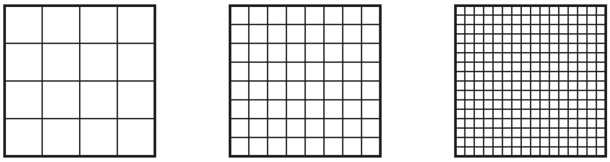


FIG. 7.1. *The three coarsest grids with $n_0 = 9, n_1 = 49$, and $n_2 = 225$ interior nodes.*

Table 7.1
*The relative discretization error* $\|\mathfrak{x}_\ell - \mathfrak{x}_9\|/\|\mathfrak{x}_9\|$ *in the* $L^2$*-norm.*

| $\ell = 3$ $n = 31^2$ | $\ell = 4$ $n = 63^2$ | $\ell = 5$ $n = 127^2$ | $\ell = 6$ $n = 255^2$ | $\ell = 7$ $n = 511^2$ | $\ell = 8$ $n = 1023^2$ |
|---|---|---|---|---|---|
| $5.1 \times 10^{-2}$ | $2.0 \times 10^{-2}$ | $7.1 \times 10^{-3}$ | $2.5 \times 10^{-3}$ | $8.9 \times 10^{-4}$ | $1.9 \times 10^{-4}$ |

Table 7.2
*The relative error* $\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\|/\|\mathfrak{x}_\ell\|$ *on level* $\ell = 6$ *in the* $L^2$*-norm for the iterates* $i = 0, \ldots, 4$.

| $i = 0$ | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ |
|---|---|---|---|---|
| $6.4 \times 10^{-3}$ | $4.2 \times 10^{-3}$ | $2.8 \times 10^{-3}$ | $2.0 \times 10^{-3}$ | $2.0 \times 10^{-3}$ |

Table 7.3
*The relative error* $\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\|/\|\mathfrak{x}_\ell\|$ *on level* $\ell = 7, 8, 9$ *in the* $L^2$*-norm for the iterates* $i = 0, \ldots, 4$.

| $\ell$ | $k$ | $i = 0$ | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ | Time |
|---|---|---|---|---|---|---|---|
| 7 | 3 | $2.3 \times 10^{-3}$ | $1.5 \times 10^{-3}$ | $9.1 \times 10^{-4}$ | $5.4 \times 10^{-4}$ | | 84.2 |
| 8 | 4 | $9.6 \times 10^{-4}$ | $6.5 \times 10^{-4}$ | $4.4 \times 10^{-4}$ | $3.0 \times 10^{-4}$ | $2.1 \times 10^{-4}$ | 1064.1 |
| 9 | 4 | $3.5 \times 10^{-4}$ | $2.4 \times 10^{-4}$ | $1.6 \times 10^{-4}$ | $1.1 \times 10^{-4}$ | $7.0 \times 10^{-5}$ | 6964.1 |

**7.1. Discretization error.** First, we perform the multigrid method with rank $k = 20$ in order to produce a reference solution on each level $\ell = 0, \ldots, 9$. The solution on level $\ell = 9$ is used to estimate the discretization error on level $\ell = 0, \ldots, 8$; see Table 7.1. The rank $k$ is large enough so that the truncation has no influence.

Alternatively, one could use the multigrid method without truncation working with full matrices instead of the $R(k)$-matrix format. For level $\ell = 3$ this takes only 2.6 seconds to compute an accurate solution, but the complexity is quadratic in $n$, so that on level $\ell = 5$ we need more than 600 seconds for the solution and on level $\ell = 8$ we would (theoretically) need approximately 775 hours.

**7.2. Truncation error.** Next, we perform the multigrid cycle on level $\ell = 6$ with fixed rank $k = 2$, so that the truncation error $\varepsilon$ due to the small rank $k$ becomes dominant; see Table 7.2. As was expected from the theory, the convergence breaks down after we get close to the solution. Since we are already at the size of the discretization error, we can stop the iteration after three steps which takes 9.7 seconds.

**7.3. Large scale problems.** The last three levels $\ell = 7, 8, 9$ in our numerical test would require storing (and computing) solution matrices $X_\ell$ of size up to $4190209 \times 4190209$. Without the low rank format the storage in double precision of these requires 128 Terabyte. In the $R(k)$-matrix format we need only 256 MB. In the following numerical example we use the $R(k)$-multigrid algorithm based on the $R(k)$-Richardson iteration. The time in seconds for the computation of a solution that is accurate up to the discretization error is given in Table 7.3.

The nested iteration combined with the multigrid method has a complexity of $\mathcal{O}(k^2 n_\ell)$ to solve the discrete system $\mathfrak{A}_\ell \mathfrak{x}_\ell = \mathfrak{b}_\ell$ on level $\ell$. Although the dependency is linear in $n_\ell$, the rank $k$ for the representation of the solution depends on $n_\ell$, typically $k = \log(n_\ell)$. Therefore, the overall complexity of our algorithm is $\mathcal{O}(n_\ell \log^2(n_\ell))$. A goal for future research is to reduce this complexity down to $\mathcal{O}(n_\ell)$.

**7.4. Second model problem.** For the second model problem from section 1.2 we consider the dependency on the parameter $\beta$ that governs the nonsymmetry of the system. For $\beta = 0$ the model problem is identical to the one considered in the previous section.

Table 7.4

*The first $k = 1, \dots, 9$ singular values of the solution $X_\ell$ on level $\ell = 5$ for the parameter $\beta \in \{1, 10, 100\}$.*

| $k$ | $\beta = 1$ | $\beta = 10$ | $\beta = 100$ |
|---|---|---|---|
| 1 | $2.6 \times 10^{-2}$ | $6.3 \times 10^{-2}$ | $1.0 \times 10^{-1}$ |
| 2 | $2.3 \times 10^{-3}$ | $1.2 \times 10^{-2}$ | $3.1 \times 10^{-2}$ |
| 3 | $4.2 \times 10^{-4}$ | $2.4 \times 10^{-3}$ | $1.3 \times 10^{-2}$ |
| 4 | $1.2 \times 10^{-4}$ | $6.8 \times 10^{-4}$ | $6.1 \times 10^{-3}$ |
| 5 | $3.2 \times 10^{-5}$ | $1.8 \times 10^{-4}$ | $3.0 \times 10^{-3}$ |
| 6 | $7.5 \times 10^{-6}$ | $5.2 \times 10^{-5}$ | $1.6 \times 10^{-3}$ |
| 7 | $1.9 \times 10^{-6}$ | $1.6 \times 10^{-5}$ | $8.1 \times 10^{-4}$ |
| 8 | $4.9 \times 10^{-7}$ | $5.0 \times 10^{-6}$ | $4.3 \times 10^{-4}$ |
| 9 | $1.5 \times 10^{-7}$ | $1.6 \times 10^{-6}$ | $2.2 \times 10^{-4}$ |

Table 7.5

*The relative error $\|\mathfrak{x}_\ell^i - \mathfrak{x}_\ell\| / \|\mathfrak{x}_\ell\|$ on level $\ell = 5$ in the $L^2$-norm and parameter $\beta \in \{1, 20, 40, 80\}$ (left: relative error; right: convergence rate).*

| $i$ | $\beta = 1$ | | $\beta = 20$ | | $\beta = 40$ | | $\beta = 80$ | |
|---|---|---|---|---|---|---|---|---|
| 1 | $1.3 \times 10^{-2}$ | .55 | $1.3 \times 10^{-2}$ | .44 | $1.9 \times 10^{-2}$ | .43 | $2.4 \times 10^{-2}$ | .41 |
| 2 | $7.1 \times 10^{-3}$ | .55 | $7.5 \times 10^{-3}$ | .57 | $1.0 \times 10^{-2}$ | .54 | $1.4 \times 10^{-2}$ | .58 |
| 3 | $4.1 \times 10^{-3}$ | .57 | $2.2 \times 10^{-3}$ | .29 | $3.6 \times 10^{-3}$ | .35 | $5.6 \times 10^{-3}$ | .40 |
| 4 | $2.4 \times 10^{-3}$ | .58 | $1.2 \times 10^{-3}$ | .55 | $3.0 \times 10^{-3}$ | .83 | $3.4 \times 10^{-3}$ | .60 |
| 5 | $1.4 \times 10^{-3}$ | .59 | $5.6 \times 10^{-4}$ | .46 | $8.5 \times 10^{-4}$ | .29 | $2.0 \times 10^{-3}$ | .58 |
| 6 | $8.1 \times 10^{-4}$ | .59 | $2.5 \times 10^{-4}$ | .45 | $4.5 \times 10^{-4}$ | .53 | $1.8 \times 10^{-3}$ | .91 |
| 7 | $4.8 \times 10^{-4}$ | .59 | $8.8 \times 10^{-5}$ | .35 | $2.4 \times 10^{-4}$ | .52 | $1.3 \times 10^{-3}$ | .74 |

For larger $\beta$ we face two distinct problems: First, the required rank for the accurate representation of the solution will increase, because the spectrum of the system matrix $A$ will become complex and approach the imaginary axis. Second, the convergence rate of the multigrid method will not be bounded away from 1 as $\beta \to \infty$, because the smoother (in this case $R(k)$-Richardson) is not suitable for convection-dominated problems.

In Table 7.4 we can clearly see that for $\beta = 100$ the decay of the singular values of the solution is less steep than for $\beta = 1$. In the multigrid method we use the damping parameter $\theta := 1/\|\mathfrak{A}_\ell\|_2$, but the coarsest grid level will now depend on the parameter $\beta$: For $\beta = 1$ we choose the coarsest grid level $\ell = 0$, and for $\beta = 20, 40, 80$ we take $\ell = 1, 2, 3$, so that the ratio $\beta \cdot h_\ell$ is constant on the coarsest grid. Of course, for larger values of $\beta$ the coarsest grid on which we have to solve the Sylvester equation by some other means will grow. The convergence rates of the multigrid iteration are given in Table 7.5. If either the damping parameter $\theta$ is chosen too large or the coarsest grid too coarse, then the multigrid iteration diverges.

**7.5. First model problem.** At last we consider the first model problem from section 1.1 (parameter $\kappa(\xi) = 10000$ for $\xi \in (\frac{1}{2}, 1) \times (0, 1)$), where a Riccati equation has to be solved. In each step of Newton's method (initial guess $X_{\ell-1}$ from the coarse grid) we have to solve a Lyapunov equation, which is done by using the multigrid method. Here we employ the Jacobi smoother, where the damping factor is computed as in section 5 for the coarsest grid solver on level $\ell = 1$, i.e., $\theta := 2/\|\mathrm{diag}(\mathfrak{A}_\ell)^{-1/2}\, \mathfrak{A}_\ell\, \mathrm{diag}(\mathfrak{A}_\ell)^{-1/2}\|_2$. In the first three steps of the multigrid method we use the same choice of the damping parameter. From step 4 on we use $\theta := 1/2$. The Cauchy matrix approximation $\tilde{\mathcal{C}}$ uses a rank of 1. We fix the discretization level $\ell = 5$ with $n_\ell = 16129$ degrees of freedom and a solution matrix

Table 7.6

*Convergence rates for the first $i = 1, \ldots, 10$ steps of the multigrid iteration in the Newton step $j = 1, 2, 3$, based on the Jacobi smoother (left: relative error; right: convergence rate).*

| $i$ | NS $j = 1$ | | NS $j = 2$ | | NS $j = 3$ | |
|---|---|---|---|---|---|---|
| 1 | $6.1 \times 10^{-3}$ | .13 | $7.5 \times 10^{-6}$ | .08 | $3.3 \times 10^{-7}$ | .74 |
| 2 | $7.8 \times 10^{-3}$ | .13 | $5.1 \times 10^{-6}$ | .67 | $2.4 \times 10^{-7}$ | .74 |
| 3 | $1.3 \times 10^{-4}$ | .16 | $3.8 \times 10^{-6}$ | .75 | $1.8 \times 10^{-7}$ | .74 |
| 4 | $7.4 \times 10^{-5}$ | .58 | $2.8 \times 10^{-6}$ | .74 | $1.3 \times 10^{-7}$ | .74 |
| 5 | $5.1 \times 10^{-5}$ | .69 | $2.1 \times 10^{-6}$ | .74 | $9.8 \times 10^{-8}$ | .74 |
| 6 | $3.6 \times 10^{-5}$ | .71 | $1.5 \times 10^{-6}$ | .74 | $7.3 \times 10^{-8}$ | .74 |
| 7 | $2.6 \times 10^{-5}$ | .73 | $1.1 \times 10^{-6}$ | .74 | $5.4 \times 10^{-8}$ | .74 |
| 8 | $1.9 \times 10^{-5}$ | .73 | $8.3 \times 10^{-7}$ | .74 | $4.0 \times 10^{-8}$ | .74 |
| 9 | $1.4 \times 10^{-5}$ | .73 | $6.1 \times 10^{-7}$ | .74 | $2.9 \times 10^{-8}$ | .74 |
| 10 | $1.0 \times 10^{-5}$ | .73 | $4.5 \times 10^{-7}$ | .74 | $2.1 \times 10^{-8}$ | .74 |

$X \in \mathbb{R}^{n_\ell \times n_\ell}$. The rank for the $R(k)$-multigrid algorithm is fixed to $k = 20$. We perform three Newton steps and apply $i = 10$ multigrid steps each.

By $X^j$ we denote the (almost) exact solution of the Lyapunov equation in the $j$th Newton step (NS) on level $\ell$. By $X_i^j$ we denote the $i$th iterate of the multigrid iteration in the $j$th Newton step. In Table 7.6 the relative error $\|X^j - X_i^j\|_2 / \|X^j\|_2$ is reported for the three Newton steps $j = 1, 2, 3$.

We conclude that the $R(k)$-multigrid method is well-suited for the solution of the linear matrix equations in each step of Newton's method to solve the algebraic matrix Riccati equation. The Jacobi smoother yields uniformly bounded convergence rates $\rho \approx 0.74$. If the parameter $\kappa$ is much smaller, i.e., $\kappa(\xi) = \mathcal{O}(1)$, then the convergence rate is $\rho \approx 0.52$.

## REFERENCES

[1] A. Antoulas, D. Sorensen, and Y. Zhou, *On the decay rate of Hankel singular values and related issues*, Systems Control Lett., 46 (2002), pp. 323–342.

[2] A. Antoulas and D. Sorensen, *The Sylvester equation and approximate balanced reduction*, Linear Algebra Appl., 351–352 (2002), pp. 671–700.

[3] R. H. Bartels and G. W. Stewart, *Solution of the matrix equation $AX + XB = C$*, Comm. ACM, 15 (1972), pp. 820–826.

[4] H. Banks and K. Kunisch, *The linear regulator problem for parabolic systems*, SIAM J. Control Optim., 22 (1984), pp. 684–696.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, London, 1996.

[6] L. Grasedyck, *Existence of a low rank or $\mathcal{H}$-matrix approximant to the solution of a Sylvester equation*, Numer. Linear Algebra Appl., 11 (2004), pp. 371–389.

[7] W. Hackbusch, *Multi-Grid Methods and Applications*, 2nd ed., Springer-Verlag, Berlin, 2003.

[8] W. Hackbusch, *Iterative Solution of Large Sparse Systems*, 2nd ed., Springer-Verlag, Berlin, 2003.

[9] W. Hackbusch, *Elliptic Differential Equations. Theory and Numerical Treatment*, 2nd ed., Springer-Verlag, Berlin, 2003.

[10] W. Hackbusch and S. Sauter, *Composite finite elements for the approximation of PDEs on domains with complicated micro-structures*, Numer. Math., 75 (1997), pp. 447–472.

[11] D. Kleinman, *On an iterative technique for Riccati equations computation*, IEEE Trans. Automat. Control, 13 (1968), pp. 114–115.

[12] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, New York, 1972.

[13] I. Lasiecka and R. Triggiani, *Control Theory for Partial Differential Equations: Continuous and Approximation Theories*, Cambridge University Press, Cambridge, 2000.

[14] R. Lezius and R. F. Tröltzsch, *Theoretical and numerical aspects of controlled cooling of steel profiles*, in Progress in Industrial Mathematics at ECMI94, H. Neunzert, ed., Wiley-Teubner, Leipzig, 1996, pp. 380–388.

[15] J. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.

[16] T. Penzl, *Eigenvalue decay bounds for solutions of Lyapunov equations: The symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.

[17] T. Penzl, *A Multi-Grid Method for Generalized Lyapunov Equations*, Technical report 24, SFB 393 at University Chemnitz, 1997.

[18] T. Penzl, *A cyclic low rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1401–1418.

[19] J. Ruge and K. Stüben, *Efficient solution of finite difference and finite element equations by algebraic multigrid*, in Multigrid Methods for Integral and Differential Equations, D. J. Paddon and H. Holstein, eds., Oxford University Press, NY, 1985, pp. 169–212.

[20] D. Hu and L. Reichel, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 172 (1992), pp. 283–313.

[21] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 895–926

# A HESSENBERG REDUCTION ALGORITHM FOR RANK STRUCTURED MATRICES*

STEVEN DELVAUX† AND MARC VAN BAREL†

**Abstract.** In this paper, we show how to perform the Hessenberg reduction of a rank structured matrix under unitary similarity operations in an efficient way, using the Givens-weight representation. This reduction can be used as a first step for eigenvalue computation. We also show how the algorithm can be modified to compute the bidiagonal reduction of a rank structured matrix, this latter method being a preprocessing step for computing the singular values of the matrix. For the main cases of interest, the algorithms we describe in this paper are of complexity $O((ar + bs)n^2)$, where $n$ is the matrix size, $r$ is some measure for the average rank index of the rank structure, $s$ is some measure for the bandwidth of the unstructured matrix part around the main diagonal, and $a, b \in \mathbb{R}$ are certain weighting parameters. Numerical experiments demonstrate the stability of this approach.

**Key words.** rank structured matrix, (zero-creating) Givens-weight representation, Hessenberg reduction, eigenvalue computation, singular value computation, structure inheritance

**AMS subject classifications.** 65F15, 15A21, 15A03

**DOI.** 10.1137/060658953

## 1. Introduction.

In this paper, we describe how for a rank structured matrix with given Givens-weight representation, we can efficiently compute its Hessenberg form using unitary similarity transformations. The algorithm is particularly useful when the underlying matrix is Hermitian.

Hessenberg reduction is often the first step for computing the eigenvalues of a given matrix $A \in \mathbb{C}^{n\times n}$. This process transforms the given matrix into Hessenberg form using a unitary similarity transformation $A \mapsto Q^H AQ$, which may be based on either Givens or Householder transformations [9, 18]. Since the resulting Hessenberg matrix has the same eigenvalues as the original matrix, the eigenvalue problem reduces to finding the eigenvalues of a Hessenberg matrix, for which efficient algorithms can then be applied such as the QR-algorithm [9, Chapters 7 and 8] and [14, Chapter 8].

Classical algorithms for the Hessenberg reduction of banded matrices are contained in [16, 15]. Concerning Hessenberg reduction for rank structured matrices, we may refer to [7, 13] for the case of weakly semiseparable matrices of semiseparability rank one and to [1] for semiseparable plus banded matrices of arbitrary semiseparability rank. But we should mention that these papers use what we call a *uv*-representation for representing these matrices, this latter condition implying some

---

*Received by the editors May 4, 2006; accepted for publication (in revised form) by N. Mastronardi February 20, 2007; published electronically August 31, 2007. This research was partially supported by the Research Council K. U. Leuven, project OT/05/40 (Large rank structured matrix computations), Center of Excellence: Optimization in Engineering, by the Fund for Scientific Research–Flanders (Belgium), G.0455.0 (RHPH: Riemann–Hilbert problems, random matrices and Padé–Hermite approximation), G.0423.05 (RAM: Rational modelling: optimal conditioning and stable algorithms), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The scientific responsibility rests with the authors.

http://www.siam.org/journals/simax/29-3/65895.html

†Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Steven.Delvaux@cs.kuleuven.ac.be, Marc.VanBarel@cs.kuleuven.ac.be).

intrinsic restrictions to the class of matrices for which the algorithm can be applied. In this paper, we will not assume such a restriction.

For the main cases of interest, the algorithm we describe in this paper is of complexity $O((ar + bs)n^2)$, where $n$ is the matrix size, $r$ is some measure for the average rank index of the rank structure, $s$ is some measure for the bandwidth of the unstructured matrix part around the main diagonal, and $a, b \in \mathbb{R}$ are certain weighting parameters.

Rank structured matrices arise in a variety of ways in the literature. For example, they arise in the study of time-varying linear systems [6]. Another example is the hierarchical $\mathcal{H}$- and $\mathcal{H}^2$-matrices of Hackbusch, Khoromskij, and Sauter [11], which appear from the discretization of integral equations. Rank structured matrices can also be successfully used for the numerical approximation of Toeplitz matrices [12].

This paper is organized as follows. In section 2, we review some topics of the Givens-weight representation from [3], paying particular attention to the so-called zero-creating Givens-weight representation. Section 3 handles the Hessenberg reduction algorithm for a lower rank structured matrix. This section contains both a theoretical part concerning structure inheritance, as well as a practical part concerning the exploitation of these inheritance results. Section 4 deals with some variants of the algorithm, showing how to exploit symmetry or general rank structure in the upper triangular part during the algorithm. We also explain here how the algorithm can be modified to compute the bidiagonal reduction of a given rank structured matrix. Section 5 deals with some numerical experiments.

## 2. Givens-weight representation.

Before proceeding to the algorithm, in this section we provide and recall some topics concerning the Givens-weight representation.

### 2.1. General definitions.

In this subsection we review the basic ideas of the Givens-weight representation from [3]. This subsection is a summary of the ideas in [3], except that Definition 1 is more general here. Readers familiar with these ideas might wish to move directly to section 2.2.

First, we define the class of rank structured matrices.

DEFINITION 1 (see [2]). [illegible] rank structure $\mathcal{R}$ [illegible] $\mathbb{C}^{m\times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_k$ [illegible] $\mathcal{B}_k$ [illegible] 4 [illegible]

$$\mathcal{B}_k = (i_k, j_k, r_k, \lambda_k),$$

[illegible] $i_k$ [illegible] $j_k$ [illegible] $r_k$ [illegible] $\lambda_k \in \mathbb{C}$ [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible] $\mathcal{R}$ [illegible] $k$

$$\operatorname{rank} A_k(i_k : m, 1 : j_k) \leq r_k, \qquad [illegible] \qquad A_k = A - \lambda_k I.$$

[illegible] $\lambda_k$ [illegible]

[illegible] $\mathcal{B}_k$ [illegible] pure [illegible] $\mathcal{B}_{\text{pure},k}$

Figure 2.1 shows an example with two structure blocks.

In practice, it often happens that the block [illegible] triangular part is also rank structured, i.e., that the matrix $A^T$ also satisfies rank structure in the sense of Definition 1. We will indiscriminately use the term [illegible] also in this case.

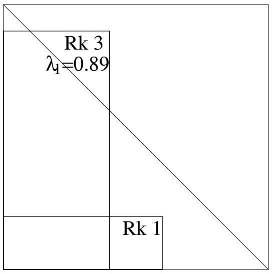


FIG. 2.1. *Example of a rank structure with two structure blocks. The left structure block* $\mathcal{B}_1$ *intersects the diagonal and has shift* $\lambda_1 = 0.89$*, while the second structure block* $\mathcal{B}_2$ *is "pure." The notation "*Rk $r$*" denotes that the structure block is of rank at most* $r$*.*

We will assume in what follows that we are working with a rank structure $\mathcal{R}$ for which there are no structure blocks that are "contained" in each other, i.e., for which the structure blocks $\mathcal{B}_k$ can be ordered such that both their row and column indices $i_k$ and $j_k$ increase in a strictly monotonic way.

Now we will try to indicate the underlying ideas of unitary-weight representations, following [3]. To this end we will take the structure in Figure 2.2 as a didactical example. First, it may be noted that this figure does not show the surrounding matrix box anymore: this reflects the fact that only the area spanned by the structure blocks will be relevant for the representation, and that the "outside world" will be inaccessible.

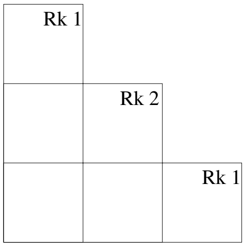


FIG. 2.2. *Example of a rank structure with three structure blocks* $\mathcal{B}_1, \mathcal{B}_2$*, and* $\mathcal{B}_3$*. We will use this example to explain the mechanism of the unitary-weight and Givens-weight representation during the following paragraphs. From now on the surrounding matrix box, as in Figure* 2.1*, will not be shown anymore.*

In what follows, we will often work with [illegible]. These are defined as unitary matrices having a block diagonal form $U = I_a \oplus Q \oplus I_b$, where $I_a, I_b$ denote identity matrices of suitable sizes. When such a unitary operation $U$ acts on the rows of a given matrix, we will represent it in a pictorial way by a vertical line segment, placed on the position of the rows on which it acts (see further).

The unitary-weight representation is obtained by reducing the structure blocks into blocks of zeros by the use of unitary row transformations. First, we apply an (elementary) unitary transformation to transform the bottom Rk 1 block into a block of zeros with one row less; see Figure 2.3.

Note that this unitary transformation acts only on the columns on the left of the vertical line which is indicated in boldface in the figure: we say that this line borders the [illegible] of the unitary transformation. Thus the action radius of the current unitary transformation is equal to 9.

Having applied this operation, note that in columns 7, 8, and 9 we have already reached the "top" of the structure. Therefore, this is now the right moment to consider the top elements of these columns and to store them. These elements will be called [illegible] and they are visualized on a grey background in Figure 2.3.

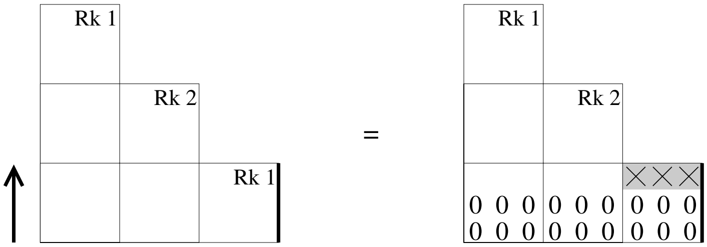


Fig. 2.3. *We apply a unitary transformation to transform the bottom two rows of the structure into zeros. This transformation acts only on the columns on the left of the vertical line which is indicated in boldface in the figure; we say that this line borders the* action radius *of the unitary transformation. Having performed this unitary transformation, the elements indicated on a grey background are stored and are called* weights.

From now on we consider columns 7, 8, and 9 as finished, and we restrict our perspective to the previous columns. We can then apply a unitary transformation to transform the middle Rk 2 block into a block of zeros, with two rows less; see Figure 2.4.

Note that again, this unitary operation acts only on the columns on the left of the vertical line indicated in boldface in the figure. Thus the action radius of the current unitary transformation is equal to 6.

Having applied this operation, note that also in columns 4, 5, and 6 we have reached the top of the structure. Therefore, this is now the right moment to consider the top elements of these columns and to store them. This yields us a second block of weights, which is again visualized on a grey background in Figure 2.4.

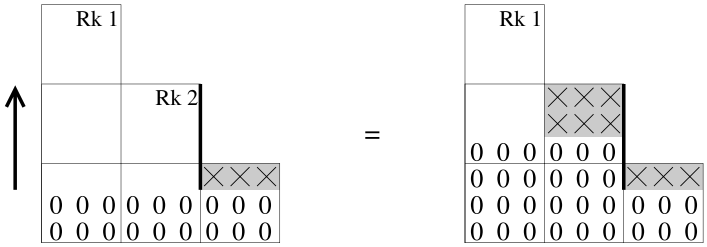


Fig. 2.4. *We apply the next unitary transformation and store the new block of weights.*

From now on we drop columns 4, 5, and 6 from our perspective. We can then apply a unitary transformation to transform the top Rk 1 block into a block of zeros with one row less; see Figure 2.5. We conclude by storing the final block of weights.

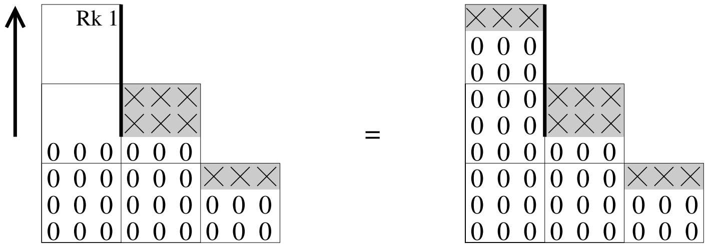


FIG. 2.5. *We apply the final unitary transformation and store the new block of weights.*

The weights can now be collected into a single matrix, which we call the [illegible]. Together with the computed unitary transformations, this matrix yields us the complete [illegible] of the given matrix; see Figure 2.6.

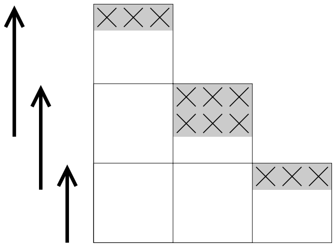


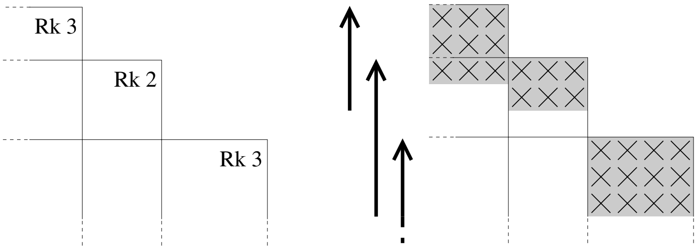
FIG. 2.6. *Schematic picture of the unitary-weight representation for the rank structure in Figure* 2.2.

Of course, to be a useful representation, the unitary-weight representation should allow the possibility to restore the original matrix which we started from. This can be done by reversing the previous steps. This reversal process is called [illegible] the unitary-weight representation and is described in [3].

Now we can come to the general definition of unitary-weight representations.

DEFINITION 2 (index sets). [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_{k=1}^K$ [illegible] $I_k = \{i_k, \ldots, i_{k+1} - 1\}$ $I_{k,\mathrm{top}} = \{i_k, \ldots, i_k + r_k - 1\}$ [illegible] $J_k = \{j_{k-1} + 1, \ldots, j_k\}$ [illegible] $k = 1, \ldots, K$ [illegible] $i_{K+1} := N + 1$ [illegible] $j_0 := 0$ [illegible] $r_{K+1} := 0$

DEFINITION 3 (unitary-weight representation). [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_{k=1}^K$ [illegible] unitary-weight representation [illegible] $A$ [illegible] $\mathcal{R}$ [illegible] $(\{U_k\}_{k=1}^K, W)$. [illegible] $U_k$ [illegible] $I_k \cup I_{k+1,\mathrm{top}}$ [illegible] $\bigcup_{l=1}^k J_l$ [illegible] $I_{k,\mathrm{top}}$ [illegible] $U_k$ [illegible] $k = K, K-1, \ldots, 1$ [illegible] $W \in \mathbb{C}^{m\times n}$ [illegible] weight matrix [illegible] $I_{k,\mathrm{top}}$ [illegible] $J_k$ [illegible] $U_k$ [illegible] 2.7



(a) (b)

FIG. 2.7. *For the rank structure in the left picture, the right figure shows a schematic picture of the unitary-weight representation.*

We can now specify from unitary-weight to Givens-weight representations. In what follows, we will use the term [illegible] to denote an elementary unitary operation which differs from the identity matrix only in two subsequent rows and columns $i$ and $i+1$. This transformation will sometimes be denoted as $G_{i,i+1}$, and the index $i$ will be called the [illegible] of the Givens transformation. Similar to our notation for elementary unitary operations, we will graphically denote the Givens transformation $G_{i,i+1}$ by means of a vertical line segment, with the height at which this line segment is standing in the figure determined by the row index $i$ (see further).

Rather than individual Givens transformations, it will be useful to work with [illegible]: these are defined as products of the form $G_{i+k,i+k+1} \dots G_{i,i+1}$, for some $k \geq 0$. Graphically, this can be considered as a collection of Givens transformations where each Givens transformation is situated precisely one position below the previous one; see Figure 2.8.

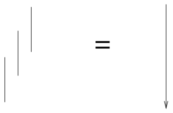


FIG. 2.8. *A Givens arrow $G_{i+2,i+3}G_{i+1,i+2}G_{i,i+1}$ consisting of three Givens transformations. Concerning this figure, we remind the reader that we consider each Givens transformation as "acting" on the rows of an (invisible) matrix standing on the right of it, and hence that the Givens transformations in the figure should be evaluated from right to left, hereby explaining the downward direction of the Givens arrow.*

The number of Givens transformations of which a Givens arrow consists will be called the [illegible] of the Givens arrow. Moreover, we define the [illegible] and the [illegible] of the Givens arrow to be the largest and the smallest row index of the Givens transformations of which the Givens arrow consists, respectively. These notions have an obvious graphical interpretation.

DEFINITION 4 (Givens-weight representation; see [3]). [illegible] $A \in \mathbb{C}^{m \times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}$ [illegible] Givens-weight representation [illegible] $A$ [illegible] $\mathcal{R}$ [illegible] $U_k$ [illegible]

- [illegible] $r_k$

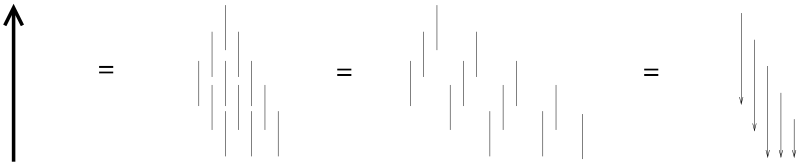


FIG. 2.9. *Suppose that the current structure block is* Rk 3*, and that the corresponding unitary transformation $U_k$ spans over six rows. Then we assume for this unitary transformation a decomposition into a product of Givens arrows of width at most* 3.

- [illegible] $U_k$ [illegible]

[illegible] 2.9

We should still explain why the assumption is made that each Givens arrow in the decomposition of $U_k$ has width at most $r_k$. To this end, recall that the unitary transformation $U_k$ serves to create zeros in a certain $\mathrm{Rk}(r_k)$ submatrix, except for its top $r_k$ rows. This effect can always be realized by a succession of Givens arrows as prescribed; see [3, section 3] for more details.

Note that by decomposing each unitary transformation $U_k$ as specified in Definition 4, we formally obtain a decomposition into a product of [illegible] Givens transformations, in the sense that the beginning and trailing Givens transformations of two subsequent unitary transformations $U_k$ may overlap. But we will not be concerned about this here, since we will work with a very special kind of Givens-weight representation to be described next.

**2.2. Zero-creating Givens-weight representation.** The algorithm to be described in this paper requires a very special kind of Givens-weight representation. Namely, we must assume it to be in "zero-creating" form, which means, loosely speaking, that each Givens transformation has to create a zero in the matrix. If this condition is not satisfied yet, then it has to be imposed, and the way to do this will be the subject of the present subsection.

DEFINITION 5 (zero-creating Givens-weight representation). [illegible] 4 [illegible] zero-creating [illegible]

(i) [illegible] $U_k$ [illegible]

(ii) [illegible] $W_{k-1}, W_k$ [illegible] $W$ [illegible] $r_{k-1} < r_k$ [illegible] $r_k - r_{k-1}$ [illegible] $W_k$ [illegible] $k = 1, \dots, K$

[illegible] $\mathcal{B}_0 : (i, j, r) = (1, 0, 0)$ [illegible]

Note that condition (i) in Definition 5 implies that the Givens transformations can be stored in a 2-dimensional array in a natural way, by storing at the $(i, j)$th block entry of this array, the $j$th Givens transformation of the Givens arrow with tail index $i$ (which must then be unique). This property is convenient for implementation purposes.

On the other hand, condition (ii) in Definition 5 might seem strange at first, but this turns out to be nothing but a natural consequence of the zero-creating character. Note that this condition implies, in particular, that the top weight block $W_1$ must be upper triangular. In fact, we will need only a weaker version of this condition.

DEFINITION 6 (weakly zero-creating Givens-weight representation). [illegible] 5 [illegible] (ii) [illegible] (ii') [illegible] $W_1$ [illegible] weakly zero-creating [illegible]

Let us transform a given Givens-weight representation into (weakly) zero-creating form. Part of this process has already been sketched in [3], but we will provide here a detailed description. First we recall the pull-through lemma from [3].

LEMMA 7 (pull-through lemma). [illegible] 3 [illegible] 3 [illegible] $Q$ [illegible]

$$Q = G'_{1,2}G_{2,3}G_{1,2},$$

[illegible]

$$Q = \tilde{G}'_{2,3}\tilde{G}_{1,2}\tilde{G}_{2,3}.$$

[illegible] 2.10

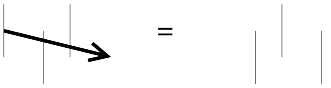


FIG. 2.10. *Pull-through lemma applied in the downward direction. One could imagine that the leftmost Givens transformation is really "pulled through" the two rightmost Givens transformations.*

Now in order to transform an arbitrary Givens-weight representation into a zero-creating one, we start on top of the matrix. We are concerned with condition (ii), and so we apply extra Givens transformations to bring the top weight block $W_1$ into upper triangular form. These Givens transformations are then simply "concatenated" to the top unitary transformation $U_1$. Next, we restore condition (i) by applying the pull-through lemma a maximal number of times in the downward direction inside this unitary transformation $U_1$; see Figure 2.11.

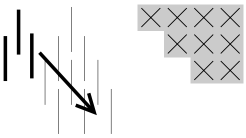


FIG. 2.11. *After applying Givens transformations to bring the top weight block in upper triangular form (indicated by the thick black lines), we apply pull-through operations to bring the top unitary operation $U_1$ of the Givens-weight representation to zero-creating form.*

Note that these operations have led to the top unitary operation $U_1$ being in zero-creating form. We can then split $U_1 = U_{1,\text{new}}U_{1,2}$, where $U_{1,2}$ consists of the bottom $r_2 - 1$ Givens arrows, i.e., those which act entirely on the rows of the next weight block $W_2$. These Givens transformations are then "transferred" to the unitary operation $U_2$ by enlarging their action radius. This means that these transformations are applied to all the columns lying between their current and their future radius, as indicated by the thick black vertical lines in Figure 2.12.

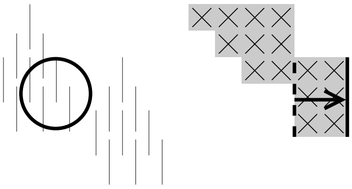


FIG. 2.12. *Having brought $U_1$ into zero-creating form in Figure* 2.11*, its bottom Givens arrows (which are encircled by the thick black line) can be transferred to the next unitary operation $U_2$ by enlarging their action radius as indicated.*

Suppose then that in the reduction process to zero-creating form, we have arrived at the unitary transformation $U_k$ for certain $2 \leq k \leq K$. We can then do the same as before: first, we apply extra Givens transformations to bring the bottom $r_k - r_{k-1}$ rows of the weight block $W_k$ into upper triangular form, provided that this number is greater than or equal to two. These Givens transformations are then just concatenated to the unitary transformation $U_k$; this is valid since it can be shown that for these bottom $r_k - r_{k-1}$ rows, there is no overlap with the Givens transformations of the unitary operations $U_{k-1}, \dots, U_1$ above. We can then again restore condition (i) by applying the pull-through lemma a maximal number of times in the downward direction inside this unitary transformation $U_k$. We can then split $U_k = U_{k,\text{new}} U_{k,k+1}$, as before and so on.

At the end of this process we will obtain a Givens-weight representation in zero-creating form. A piece of this is shown in Figure 2.13. Note that the tails of the subsequent Givens arrows in this figure (indicated by the thick black lines) are indeed strictly monotonically proceeding upwards.

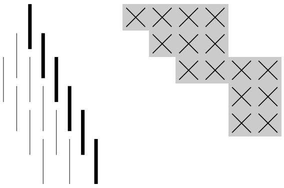


FIG. 2.13. *Part of a zero-creating Givens-weight representation.*

To explain the name zero-creating, it can be shown that each of the Givens transformations of a zero-creating Givens-weight representation has to create a zero in an appropriate place of the matrix. We will need only a weaker version of this result.

LEMMA 8 (weakly zero-creating). [illegible] $n, n-1, \dots, i_1 + 1$ [illegible]

[illegible]. Suppose that we compress the given matrix by means of the unitary operations of the Givens-weight representation. At the moment that we apply the tail of the first Givens arrow $G_{k,k+1}$, we can apply the remaining Givens transformations of which the Givens arrow consists. But instead of doing this, we can already apply the tail of the next Givens arrow $G_{k-1,k}$, because this operation acts on rows $k-1, k$, which are strictly disjoint from the rows $k+1, k+2, \dots$ on which the remaining part of the current Givens arrow acts. Repeating this argument, we can rearrange the compressing unitary operations as $V_2V_1$, where $V_1$ consists of the succession of all the subsequent tails of the Givens arrows and $V_2$ contains the remaining parts of the Givens arrows. Since the weakly zero-creating character implies that the first column of the top weight block $W_1$ must be in upper triangular form, and because $V_2$ acts only on rows strictly below the top nonzero entry of this column, we see that these zeros in the first column must have been created purely by means of the [illegible] of the subsequent Givens arrows. ☐

This lemma states that the tails of the Givens arrows, which are indicated by the thick black lines in Figure 2.13, have to create zeros in the first column of the underlying matrix. Both the meaning and the proof of this lemma can be illuminated by interpreting them in terms of this figure.

A similar property holds for the actual zero-creating Givens-weight representation. The idea is that the corresponding Givens transformations must create zeros in the subsequent columns of the matrix. Due to the presence of the low rank blocks, extra zeros will be induced automatically in the further columns during this process. We can then look further and create zeros in the first column which has not been zeroed out yet. It is possible to give an exact expression for the subsequent places where the zeros have to be created, but this expression is rather complicated and we believe that it does not provide extra insight. Moreover, we will not need to know these details for this paper.

Finally, we describe here an alternative construction of zero-creating Givens-weight representations. Since this construction will not be used in the remainder of this paper, the reader may skip the following paragraphs.

The starting point is that we must possess at least a "dual" Givens-weight representation, which we describe now.

Definition 9 (rank-decreasing). [illegible] $U_k$ [illegible]

Note that this definition is the same as in condition (i) of Definition 5 above, except that the word tails has been replaced by tops. Such Givens-weight representations were called [illegible] 2 in [3].

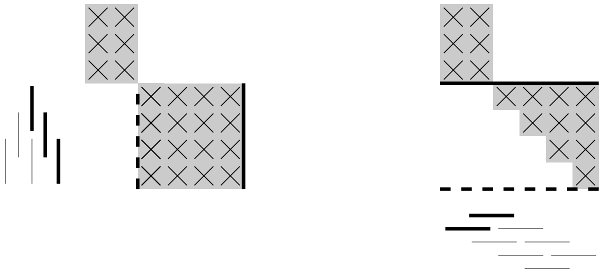
Just as zero-creating Givens-weight representations are "efficient" in the sense that each Givens transformation effectively creates a zero in the matrix, the efficiency of the rank-decreasing version consists of the fact that each Givens arrow definitively eliminates a row of the matrix, which is not touched by the next Givens arrows anymore, due to the monotonicity of the tops of the Givens arrows. The reader should try to see this. Moreover, this feature is important since it allows rank-decreasing Givens-weight representations to arise in several, natural ways; see [3].

We recall here the general fact that Givens-weight representations can also be constructed by means of unitary [illegible] instead of row operations. In particular, the rank-decreasing concept can be translated to the case of a column-based Givens-weight representation as well, by replacing in Definition 9 the word upwards by rightwards.

Concerning this last paragraph, we recall that there has been described a so-called [illegible] in [3] to transform a column-based into a row-based Givens weight representation, or vice versa. This process will be illustrated in a moment. Let us now motivate our interest in this process.

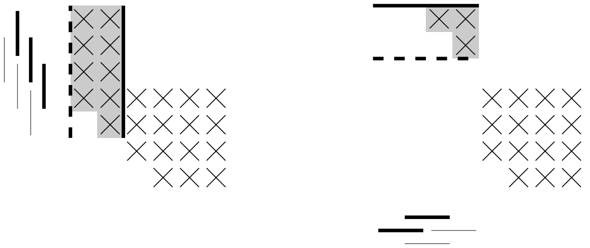
Theorem 10 (duality theorem). [illegible] [3] [illegible]

[illegible] We consider here the case of a rank-decreasing, column-based Givens-weight representation. The swapping process is illustrated in Figure 2.14.



(a) Apply row operations to compress the bottom weight block $W_K$.

(b) Spread out $W_K$ by means of the rank-decreasing unitary operation $U_K^H$.



(c) Apply further row operations to compress the next weight block.

(d) We can now go on to spread out by means of the next rank-decreasing unitary operation $U_{K-1}^H$.

FIG. 2.14. *Swapping from a rank-decreasing, column-based into a zero-creating, row-based Givens-weight representation.*

Let us comment on this figure. We start by bringing the bottom weight block $W_K$ in upper triangular form by a series of auxiliary row operations. We can obviously do this in a “zero-creating” way, a fact which is indicated by highlighting the tails of the subsequent Givens arrows in Figure 2.14(a); see also Figure 2.15.

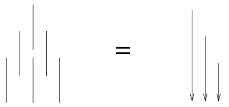


FIG. 2.15. *Zero-creating character of Figure* 2.14(a).

Next, we spread out the bottom weight block $W_K$ by applying the “decompressing" unitary operation $U_K^H$ to it. This operation $U_K^H$ is shown in Figure 2.14(b), where we assume that the subsequent decompressing Givens transformations have to be applied from top to bottom. Since we assumed the given Givens-weight representation to be rank-decreasing, it can be checked that the operation $U_K^H$ is such that its leftmost Givens transformations, highlighted in Figure 2.14(b), monotonically proceed leftwards. This property implies the crucial observation that the upper triangularity of the current weight block is [illegible] during the spreading-out process.

In particular, this preservation of upper triangularity implies that condition (ii) in the definition of zero-creating Givens-weight representation holds, i.e., when $r_k > r_{k-1}$, then the $r_k - r_{k-1}$ bottommost rows of the swapped weight block $W_k$ are upper triangular; see, e.g., the vanishing of the (7,3) element in Figure 2.14(c).

We have now arrived at the next weight block $W_{K-1}$. We compress this weight block by means of some new auxiliary row operations. By the fact that the upper triangularity of the previous weights was preserved, we can do this by just continuing the zero-creating pattern of Givens transformations on the rows; see Figure 2.14(c).

If follows from the above observations that the swapped Givens-weight representation must indeed be zero-creating. The final weight matrix is shown in Figure 2.16.

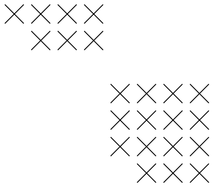


Fig. 2.16. *Final weight matrix of the resulting zero-creating, row-based Givens-weight representation at the end of the swapping process.*

Conversely, we can swap from a zero-creating, row-based into a rank-decreasing, column-based Givens-weight representation. Since this process is "dual" to the one above, it will not be shown anymore. □

Summarizing, we have described two ways of constructing (weakly) zero-creating Givens-weight representations, namely, either by the use of [illegible] techniques, as in Figures 2.11 and 2.12, starting from an arbitrary Givens-weight representation, or by the use of [illegible] techniques, as in Figure 2.14, assuming that a rank-decreasing Givens-weight representation is available. The pull-through scheme will be the one that we use in practice.

In the next section, we will come to the main theme of this paper.

## 3. Hessenberg reduction.

Assuming now that we have a weakly zero-creating Givens-weight representation to our disposal, we can start the reduction algorithm into Hessenberg form.

### 3.1. Structure propagation.

First, we will consider the structures which are propagated during the reduction of a matrix into Hessenberg form. First, by the fact that we use only unitary similarity transformations, it is easy to see that the properties of being Hermitian plus low rank, unitary plus low rank, and so on will be preserved by the reduction process. This corresponds to what were called "polynomial structures" in [2].

Still, following the analogy with [2], we would also like rank structures to be preserved during the Hessenberg reduction process. To this end, we have to make the assumption that the reduction process is based on Givens transformations, i.e., that in the $k$th step of this process, $k = 1, \ldots, n-1$, the $k$th column of the matrix is brought in Hessenberg form by an upwards pointing sequence of Givens transformations $G_{n-1,n}, \ldots, G_{k+1,k+2}$. In order to preserve the eigenvalue spectrum, we then multiply with the Hermitian transposes of these Givens transformations $G^H_{n-1,n}, \ldots, G^H_{k+1,k+2}$ to the columns, hereby not destroying the created zeros anymore. This process is illustrated by a 4 by 4 example in Figure 3.1.

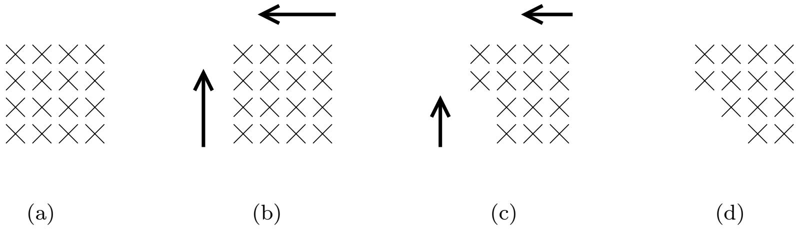

(a) (b) (c) (d)

FIG. 3.1. *Hessenberg reduction for a 4 by 4 matrix.*

Clearly, it will be sufficient if we can characterize the structure propagation after the Hessenberg reduction of the [illegible] column.

THEOREM 11 (structure propagation). [illegible] $A \in \mathbb{C}^{n\times n}$ [illegible] $\mathcal{B} = (i, j, r, \lambda)$ [illegible] $i \geq 2$ [illegible] $A|_{\mathcal{B}}$ [illegible] $A|_{\mathcal{B}}$ [illegible] $A$ [illegible] $\mathcal{B}$ [illegible] $A$ [illegible] $\mathcal{B}$ [illegible] $\tilde{\mathcal{B}} = (i+1, j+1, r, \lambda)$. [illegible] 3.2

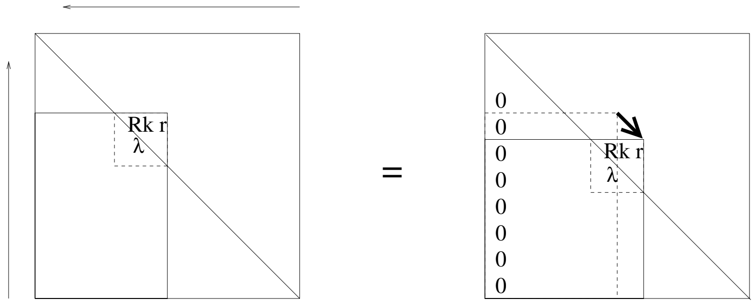


FIG. 3.2. *Structure propagation after the Hessenberg reduction of the first column.*

[illegible]. First, we show that the proof can be reduced to the case of a [illegible] structure block. To this end, note that by definition, the matrix $A_{\text{pure}} := A - \lambda I$ satisfies the pure structure block $\mathcal{B}_{\text{pure}}$ which is obtained from $\mathcal{B}$ by putting the shift element equal to zero. Now let us denote with $Q^H$ the unitary row transformation which reduces the first column of $A$ into Hessenberg form. Since $Q^H$ does not involve the top row of the matrix, it must also reduce the first column of $A_{\text{pure}} = A - \lambda I$ into Hessenberg form. Moreover, it holds that $Q^H AQ = Q^H A_{\text{pure}}Q + Q^H(\lambda I)Q = Q^H A_{\text{pure}}Q + \lambda I$. Hence indeed, it will suffice to prove the theorem for the matrix $A_{\text{pure}}$.

To prove this remaining case, we use an easy argument based directly on the Givens transformations, as shown in Figure 3.3.

Let us comment on this figure. The figure starts by applying Givens transformations to the rows, in order to make the first column Hessenberg. Thus at the moment that we are at the point of "leaving" the structure block $\mathcal{B}$, the first column of $A|_{\mathcal{B}}$ will be entirely zero, except for its top element, which is nonzero by the assumptions in the theorem. The key observation is then that the top row of $A|_{\mathcal{B}}$ cannot be written as a linear combination of the further rows, and hence the structure block obtained by removing this top row from $\mathcal{B}$ must be of rank at most $r - 1$; see Figures 3.3(a) and 3.3(b).

The proof can now be finished by remarking that this $\text{Rk}(r-1)$ structure block cannot be destroyed anymore by the next row operations. On the other hand, the application of the column operations will result in the structure block being enlarged by one column and with rank increased by one; see Figures 3.3(c) and 3.3(d). ☐

It should be noted that the above theorem holds only under the condition that the first column of $A|_{\mathcal{B}}$ is nonzero. Although this condition is very mild, it excludes the case of rank zero structure blocks; in this case, one should work with the top induced rank one structure block instead; see Figure 3.4.

Let us note some more topics concerning the above proof.

[illegible] 12.

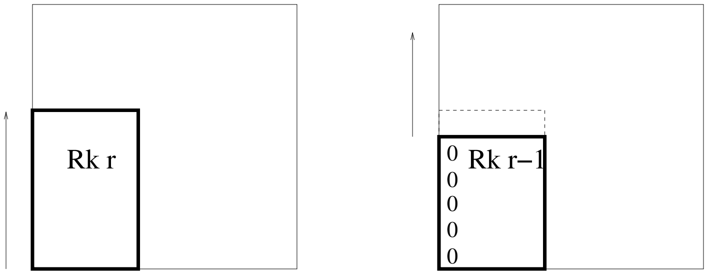
1. By repeating the argument in Figures 3.3(a) and 3.3(b) several times, it follows that the given structure block can be transformed into a new structure block with $k$ rows less, and with rank diminished by $k$, by means of a sequence of Givens arrows of width $k$ creating zeros in the first $k$ columns; see also [4].
2. It is clear that in finite precision arithmetic, creating zeros in the first column and then expecting the rank to decrease, as in Figures 3.3(a) and 3.3(b), is likely to be an unstable procedure. This is because if the first column is close to the zero vector, then it can severely deviate from the overall column space of the low rank block, due to round-off errors. The solution to this problem will be to compute the zero-creating Givens-weight representation by means of the [illegible] techniques which were described in section 2.2.

Although Theorem 11 was stated for general shift elements $\lambda$, we will first exploit it for [illegible] rank structures in what follows. This process will be described in the next subsection.

**3.2. Hessenberg reduction algorithm.** In this subsection we describe the algorithm for the Hessenberg reduction of the given rank structured matrix. We start with an algorithm for bringing the first column into Hessenberg form. The upper triangular part of the matrix will currently be assumed to be unstructured.

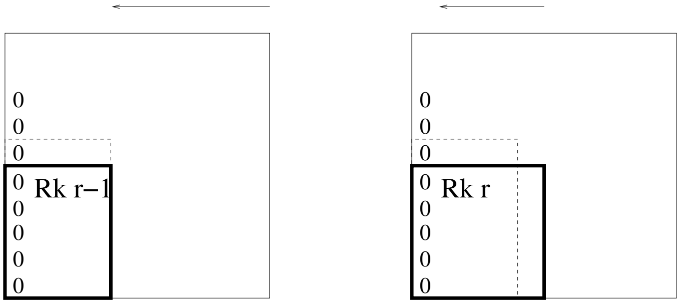
Note that the Givens transformations that bring the first column in Hessenberg form have already been [illegible], precisely by the concept of weakly zero-creating Givens-weight representation. Indeed, Lemma 8 guarantees that these transformations are nothing but the tails of the subsequent Givens arrows, as highlighted by the thick black lines in Figure 2.13. This means that the algorithm will suffice with "peeling off" these tails from the subsequent Givens arrows, in a sense to be made exact later.

The idea of the algorithm will be illustrated for the starting rank structure shown in Figure 3.5(a). The corresponding weakly zero-creating Givens-weight representa-



(a) Apply the first Givens transformations to the rows. They will transform the given Rk $r$ block into a Rk$(r-1)$ structure block with one row less.

(b) Apply the remaining Givens transformations to the rows.



(c) Apply the transposed Givens transformations to the columns, until the Rk$(r-1)$ structure block is reached.

(d) Enlarge the Rk$(r-1)$ block into a Rk $r$ structure block with one more column, and apply the remaining Givens transformations to the columns.

FIG. 3.3. *The figure shows the propagation of pure structure blocks by the Hessenberg reduction process directly in terms of the Givens transformations.*

tion is shown in Figure 3.5(b). Note that this figure shows a [illegible]-weight rather than a [illegible]-weight representation, in order not to overload the figure.

The application of the Givens transformations $G_{k,k+1}$ to the rows will be achieved by what we call a [illegible]. On the other hand, the application of the Givens transformations $G^H_{k,k+1}$ to the columns makes use of the general techniques for updating the Givens-weight representation under the influence of Givens transformations reported in [3], in the form of what we called there a [illegible]. These processes are shown in Figure 3.6.

Let us comment on this figure. Figure 3.6(a) shows the starting Givens-weight representation. It is assumed here that the first Givens transformations have already

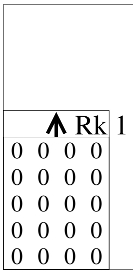
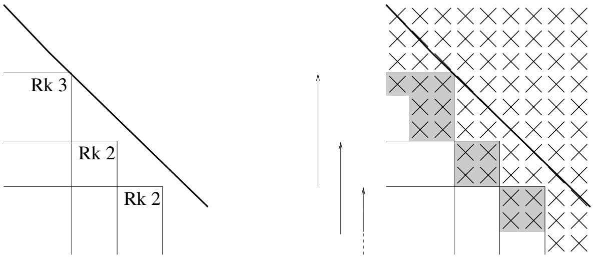


FIG. 3.4. *In case of a rank zero structure block, one should work with the top induced* Rk 1 *structure block which is indicated in the figure.*



(a) (b)

FIG. 3.5. *Starting rank structure with corresponding weakly zero-creating Givens-weight representation.*

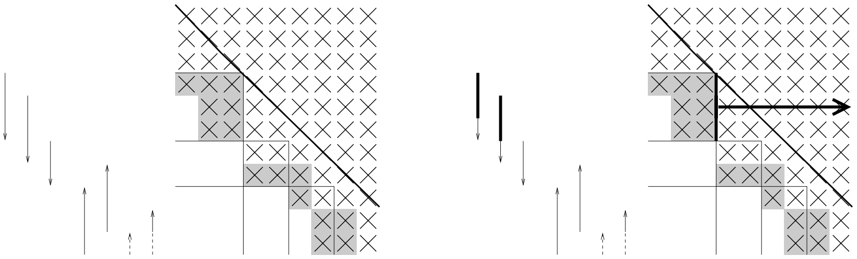
been peeled off from the structure, up to the transformation $G_{6,7}$, and that they have been applied to the rows and the columns.

The application of the [illegible] operations has led to the fact that the Givens arrows in several unitary transformations have " shrunk" because they have "lost their tail." Correspondingly, some of the originally grey elements of the two bottom structure blocks have turned white in Figure 3.6(a), since there is no Givens transformation anymore which acts upon them. We note that this peeling-off process corresponds to the Rk $r$ structure blocks turning into Rk$(r-1)$ structure blocks with one row less, as explained before.

The application of the [illegible] operations has led to the fact that the original unitary operations $U_k$ are now interlaced with some auxiliary Givens arrows which were computed during the algorithm. Furthermore, some of the elements, such as the $(9,6)$ element in Figure 3.6(a), are assumed to be disturbances coming from the application of the previous Givens transformations to the columns.

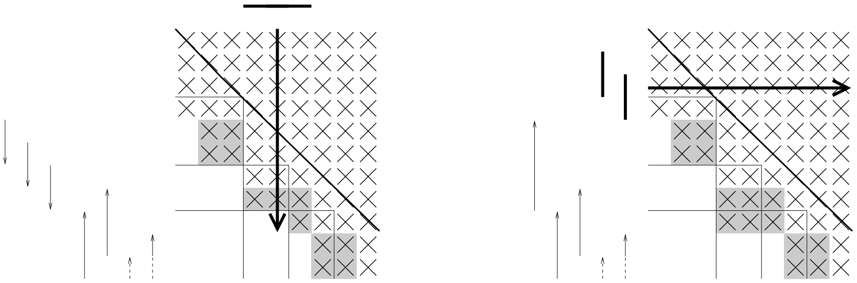
Now we are at the moment of peeling off the next Givens transformations. This will cause the corresponding Givens arrows to lose their tail. While applying these Givens transformations, we exploit the fact that the result after their application has already been partially precomputed. This is why we apply them only to the columns strictly on the right of their current action radius; see Figure 3.6(b).

Note that after their application, the top row of the top weight block will have been "completely released" in the sense that there are no Givens transformations anymore which act upon it. This means that these elements turn from grey into



(a) Starting situation. We assume that the tails of the Givens arrows up to $G_{6,7}$ have already been peeled off. Only the top unitary transformation is shown as a decomposition of Givens arrows.

(b) Peel off the tails of the next Givens arrows and apply them to the rows.



(c) Multiply the transposes of the Givens transformations of Figure 3.6(b) to the columns.

(d) Apply the final Givens transformations to the rows.

FIG. 3.6. *Hessenberg reduction of the first column (a-d).*

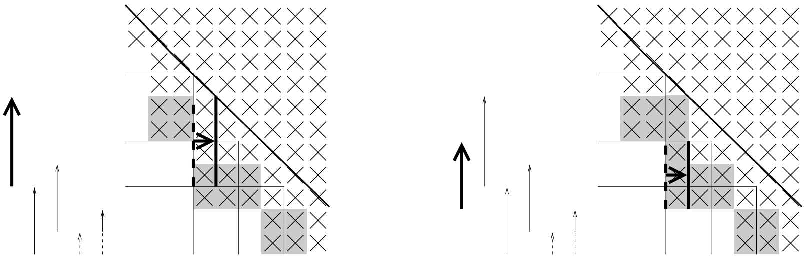
white; see Figures 3.6(b) and 3.6(c).

To complete the similarity transformation, we should apply the Hermitian transposed operations to the columns. This is done in Figure 3.6(c).

We can then go on to apply the final Givens transformations for bringing the first column in Hessenberg form; see Figure 3.6(d). We would then like to complete the similarity transformation and apply these final Givens transformations on the columns too. But this means that we are at the point of contaminating the structure block in columns 2, 3 with some of the elements in the column on the right of it. This contamination would lead to a mix of real-size elements and weights, which is definitely not allowed.

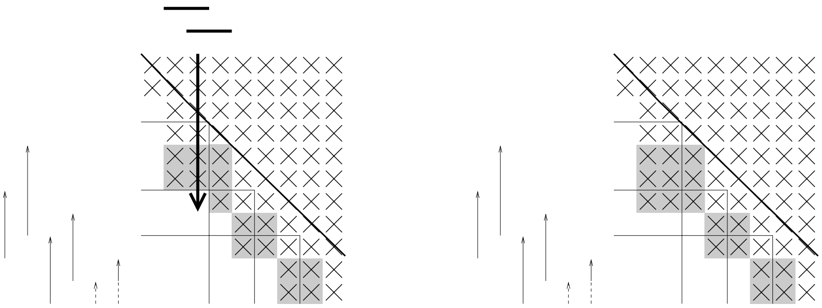
The solution consists of enlarging the action radius of the Givens-weight representation. This means that we bring the contaminating column, which is standing just on the right of the structured part, "into" the representation; see Figure 3.6(e). The corresponding elements in Figure 3.6(f) have then turned from white into grey.

Having done this, it is now safe to apply the desired Givens transformations to the columns. But we do not do this yet, since applying all these column operations would lead to a complete fill-in in the lower triangular part. We want to minimize this fill-in



(e) Enlarge the action radius of the row representation.

(f) Apply auxiliary Givens transformations to the rows to bring the weights as much as possible to the top.



(g) Multiply the transposes of the Givens transformations of Figure 3.6(d) to the columns.

(h) We have reduced the first column into Hessenberg form.

FIG. 3.6. *(cont.) Hessenberg reduction of the first column (e-h).*

as far as possible, and therefore we first apply some auxiliary Givens transformations to the rows, to bring the newly introduced weights as far as possible to the top; see Figure 3.6(f).

Having done all of these preparations, we can finally apply the desired Givens transformations to the columns; see Figure 3.6(g). We have then completely reduced the first column of the matrix into Hessenberg form.

Figure 3.6(h) shows the final Givens-weight representation. Note that the weight blocks have uniformly moved one position to the bottom right position, as predicted by Theorem 11.

Note that the algorithm has led to a Givens-weight representation in [illegible] form, i.e., a unitary-weight representation for which each unitary transformation $U_k$ has a natural decomposition of the form $U_k = \tilde{V}_k V_k$, where $V_k$ is a unitary transformation situated on "top" of $U_k$, containing what is left of the original Givens-weight representation, and $\tilde{V}_k$ is a unitary transformation situated on the "bottom" of $U_k$, containing the chain of auxiliary Givens transformations computed during the Hessenberg reduction algorithm; see Figure 3.6(h).

Now we would like to go on by making the second column Hessenberg. This means that we should first bring the Givens-weight representation back to its weakly zero-creating form, using the pull-through techniques of section 2.2. In fact, these techniques must be slightly modified since we are working here with a Givens-weight representation in [illegible] form, which is not a Givens-weight representation in the strict sense anymore.

Let us briefly apply these techniques to the present situation. First, we apply an extra upwards pointing Givens arrow, in order to restore the upper triangularity of the first column of the top left weight block in Figure 3.6(h); see Figure 3.7.

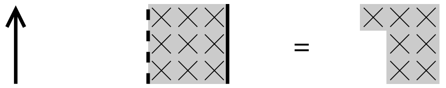


FIG. 3.7. *We apply an extra unitary operation to bring the first column of the top weight block in Figure* 3.6(h) *in upper triangular form.*

The resulting unitary transformations, to which this extra Givens arrow is added, are shown in Figure 3.8.

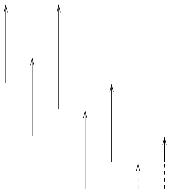
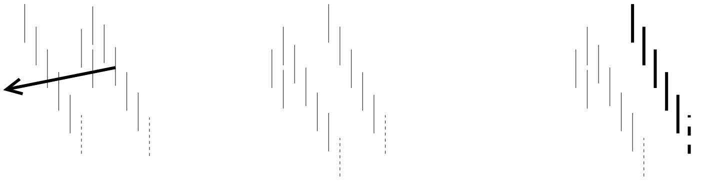


FIG. 3.8. *Resulting unitary transformations of Figure* 3.6(h).

Next, we apply the pull-through lemma a maximal number of times in the downward direction, in order to bring the representation back into (weakly) zero-creating form; see Figure 3.9.

While the pull-through process proceeds from top to bottom of the matrix, one should not forget to appropriately enlarge the action radii of the involved Givens transformations, as explained in section 2.2. The treatment of a single step in this process is shown in Figure 3.10.



(a) (b) (c)

FIG. 3.9. *The left picture shows a decomposition of Figure* 3.8 *in terms of individual Givens transformations. It is then brought to weakly zero-creating form by repeatedly using the pull-through lemma, resulting in the situation of the middle picture.*

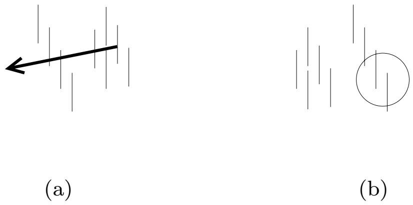

(a) (b)

FIG. 3.10. *The picture shows a specification of Figure* 3.9. *After applying the pull-through operations inside the top unitary operation* $U_1$ *as shown in the left picture, the resulting unitary operation is decomposed as* $U_1 = U_{1,\mathrm{new}} U_{1,2}$*, where the Givens transformations of* $U_{1,2}$ *are circled in the right picture. One can then enlarge the action radius of* $U_{1,2}$ *so that it is transferred to the next unitary operation* $U_2$*. This scheme can then be repeated for the next unitary operations* $U_2, U_3, \ldots$.

At the end of the pull-through process, the Givens transformations indicated by the thick black lines in Figure 3.9(c) will constitute the new tails of the Givens arrows, which will hence be peeled off during the Hessenberg reduction of the second column.

From the schematic illustration of the pull-through process in Figure 3.9, we can note the interesting fact that the (updated) chain of auxiliary Givens transformations, used for the auxiliary upward movement during the Hessenberg reduction algorithm, replaces the original tails of the Givens arrows at the end of the pull-through process; see Figure 3.9(c). Hence they will be precisely the Givens transformations to be peeled off during the Hessenberg reduction of the second column.

Finally, we have now obtained a Givens-weight representation which is back in weakly zero-creating form. We are then ready to bring the second column in Hessenberg form as well. Since this problem is completely similar to the Hessenberg reduction of the first column, it will not be explained anymore.

## 4. Some modifications to the algorithm.

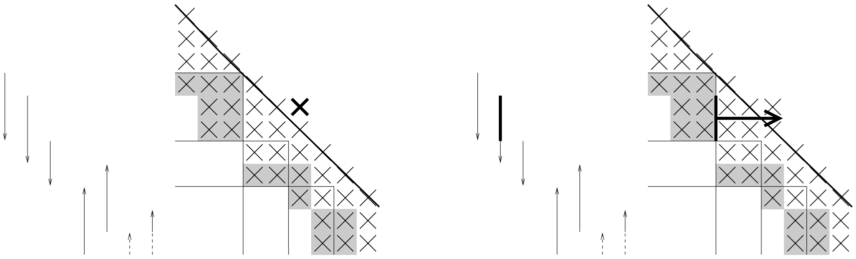
In this section we describe some variants to the Hessenberg reduction algorithm of the previous section. We start with a treatment of rank structure lying in the upper triangular part.

### 4.1. Exploiting rank structure in the upper triangular part.

In this subsection we describe how the efficiency of the algorithm can be improved by an order of magnitude in case the matrix $A$ has rank structure in its upper triangular part as well. The total algorithm complexity will decrease in this way from the cubic $O(n^3)$ to the quadratic $O((ar + bs)n^2)$, where $r$ is some measure for the average semiseparability rank, $s$ is some measure for the bandwidth of the unstructured matrix part around the main diagonal, and $a, b \in \mathbb{R}$ are certain weighting parameters.

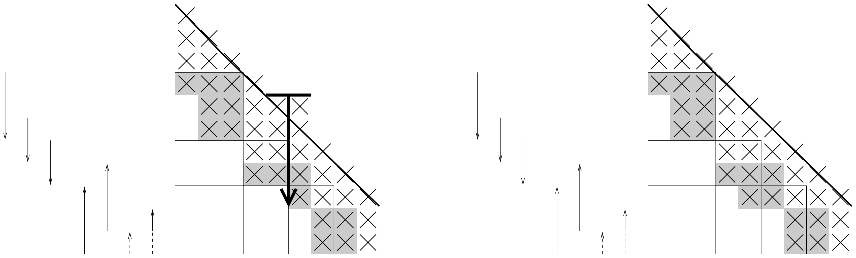
We start with the case of rank structure originating from the fact that $A$ is Hermitian, or more generally, when it is Hermitian plus a low rank correction in the sense that $A - A^H = \mathrm{Rk}\ k$ with $\mathrm{Rk}\ k$ a matrix of rank at most $k$. The algorithm can proceed then by just keeping track of the lower triangular part, since then the upper triangular part is known by symmetry. In the case where $A$ is Hermitian up to some low rank correction, one should also keep track of the low rank correction matrix $\mathrm{Rk}\ k$. The algorithm proceeds then by only computing each time the required superdiagonal element, by symmetry, next applying the row and column operation, and then again removing the superdiagonal element; see Figure 4.1.

It is easy to check that in this way, the complexity of the Hessenberg reduction



(a) Compute the required superdiagonal element, and fill it in the weight matrix.

(b) Apply the current Givens transformation to the rows.



(c) Apply the transposed Givens transformation to the columns.

(d) The current superdiagonal element has no role anymore, and therefore it is removed from the weight matrix.

FIG. 4.1. *Exploiting symmetry.*

algorithm for the $k$th column decreases to $O((ar+bs)n)$, for suitable $a, b \in \mathbb{R}$. Hence the [illegible] cost of the Hessenberg reduction algorithm reduces to $O((ar+bs)n^2)$.

Next, we describe how, even if the matrix is not Hermitian up to some low rank correction, rank structure in the upper triangular part can be exploited. To this end, the reader should reacquaint familiarity with the propagation mechanism for structure blocks in the [illegible] triangular part in Figure 3.3. It can then be noted that the argument in Figures 3.3(a) and 3.3(b) crucially depends on the fact that zeros are created in the lower triangular part of the matrix during the Hessenberg reduction process.

The corresponding propagation of rank structure in the upper triangular part does [illegible] benefit from such a creation of zeros, as shown in Figure 4.2.

It should be clear from Figure 4.2 that the ranks could soon start to increase. After a few columns have been transformed into Hessenberg form, there will probably be not much left of the rank structure, due to the growth of the rank indices of the rank structure.

There is one notable exception to this idea, namely, we know that the ranks should stay constant also in the case where $A$ is a [illegible] matrix, or unitary up to some low rank correction, because then the rank structure in the upper triangular part is a direct consequence of that in the lower triangular part; see, e.g., [5]. We will

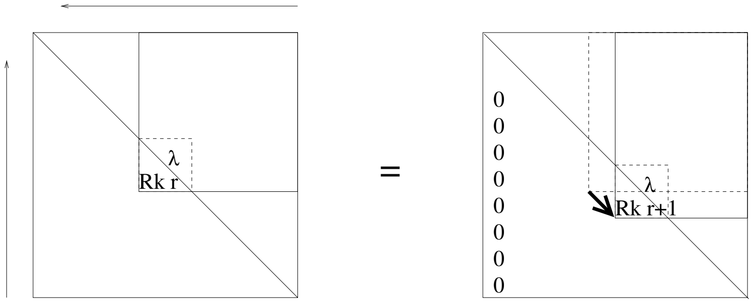


FIG. 4.2. *Structure propagation in the upper triangular part after the Hessenberg reduction of the first column. Note that the structure block moves one position to the bottom right position, but that the original rank index* $r$ *can increase to* $r + 1$.

come back to this later.

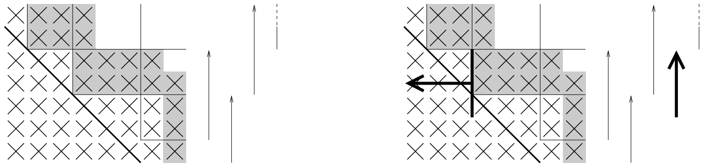
Anyway, also in the case of arbitrary rank structure in the upper triangular part, and where the ranks increase, we will give the tools to update the representation. These tools are based on the methods for updating a Givens-weight representation under the action of Givens transformations described in [3]. We saw there how to update the representation using what was called concatenation, pull-through, generalized swapping, and generalized regression techniques.

Essentially, the only difference with respect to [3] is that we want to apply the disturbing Givens transformations [illegible] on rows and columns. This will lead to two methods being applied at the same time. We consider here the case where the Givens-weight representation of the upper triangular part is based on [illegible] operations. We will then simultaneously have a mixture of concatenation (for the row operations) and generalized regression techniques (for the column operations). See Figure 4.3.

Let us comment on this figure. Figure 4.3(a) shows the starting Givens-weight representation for the structured upper triangular part. Note that we preferred here to show the [illegible] rather than the compressing unitary operations of the Givens-weight representation, i.e., in order to obtain the full matrix, we should spread out by means of the subsequent unitary operations shown in Figure 4.3(a), evaluating from left to right. The bottommost unitary transformation is assumed to be a disturbance coming from previous steps.

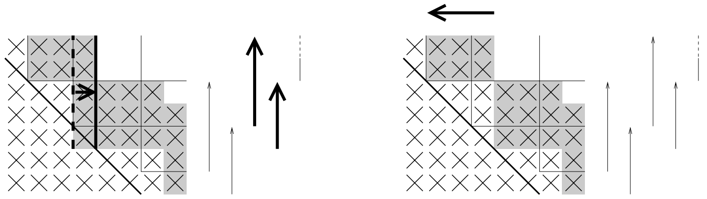
We are now at the moment of applying the next Givens transformations to the rows; see Figure 4.3(b). We apply these transformations only to columns $1, \ldots, 3$. For the application to the structured upper triangular part in columns $4, \ldots$, we do not actually apply these Givens transformations, but instead just [illegible] them to the Givens-weight representation. To see what this means, the reader should recall that the weight matrix contains a kind of "compressed" information about the full matrix. Moreover, in order to obtain the real values of these elements, we should spread out the weight matrix by applying all the subsequent unitary operations, evaluating from left to right. The concatenation process means then simply that, at the very end of this decompressing process, one should apply the newly added Givens transformations as well, so that they can be considered from now on as being part of the Givens-weight representation.

Now we would like to apply the transpose of these Givens transformations to the columns. To do this in a valid way, we have to avoid a mixture of real-size elements and weights. Therefore we first regress the action radius of the representation; see



(a) Starting Givens-weight representation. The arrows on the right denote the unitary transformations used to *spread out* the structure.

(b) Apply the next Givens transformations to columns 1,. . . ,3, and concatenate them to the representation.



(c) Regress the action radius of the representation.

(d) Apply the transposed Givens transformations of Figure 4.3(b) to the columns.

FIG. 4.3. *Updating the structured upper triangular part during the Hessenberg reduction algorithm, assuming that it is represented by a row-based Givens-weight representation.*

Figure 4.3(c). We can then safely apply the Givens transformations to the columns; see Figure 4.3(d). The next operations are not shown anymore.

Finally, let us come back to the case where the rank structured matrix is [illegible] [illegible]. In this case, the above algorithm will yield that the ranks increase from $r$ to $r+1$ in each step, although we know that they should stay constant. Thus it should be possible to perform a [illegible] of the structure after each step to approximate the ranks again by their actual values. If the structure blocks in the upper triangular part have been chosen wisely, this implies an extra term $O(r^2n)$ for the complexity in each step; see [3]. This [illegible] term in the rank index can be avoided by just performing the numerical approximation procedure [illegible] $O(r)$ [illegible] in the Hessenberg reduction process. The above $O(r^2n)$ term can then be distributed over $O(r)$ different steps, so that the total algorithm complexity becomes $O((ar+bs)n^2)$, for suitable $a, b \in \mathbb{R}$, just as in the Hermitian case.

Summarizing, we have now described how the Hessenberg reduction algorithm can be modified to take advantage of the rank structure in the upper triangular matrix part. The next subsection considers a second modification to the algorithm.

**4.2. Bidiagonal reduction algorithm.** In this subsection we will explain how a rank structured matrix can be reduced to bidiagonal form. This means that we will apply an operation of the form $A \mapsto UAV$, where $U$ and $V$ are possibly different unitary transformations. Such a reduction does not preserve the eigenvalue spectrum, but it can be used as the first step for computing the [illegible] of the given matrix [8, 9].

Before starting, we recall that during the [illegible] reduction process, the ranks in the lower triangular part were preserved, but those in the upper triangular part could increase from $r$ to $r+1$ in each step. The reason underlying this was that the Hessenberg reduction created zeros only in the lower triangular part of the matrix, implying that the argument of Figures 3.3(a) and 3.3(b) is invalid for the structure blocks in the upper triangular part.

This conclusion does not hold anymore for the bidiagonal reduction process. Indeed, by exploiting the freedom of applying possibly different unitary transformations $U$ and $V$ to rows and columns, this reduction succeeds in a creation of zeros in [illegible] the lower and upper triangular part of the matrix. Hence it is easy to see that the structure propagation argument of Figures 3.3(a) and 3.3(b) will be valid now for [illegible] the lower and the upper triangular part, without any increase of ranks, in any case. On the other hand, a small price has to be paid in the sense that the structure propagation will hold now only for [illegible] rank structures; see Figure 4.4.

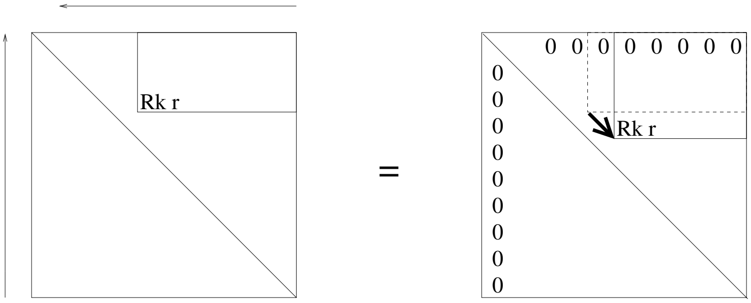


FIG. 4.4. *Structure propagation in the upper triangular part after the bidiagonal reduction of the first column and row. Note that the given, pure structure block moves one position to the bottom right corner, with rank index r being preserved. A similar result holds for the lower triangular part as well.*

Let us now discuss the practical implementation of the bidiagonal reduction process. The algorithm will closely follow the Hessenberg reduction process of section 3. The starting point will be that we have available a row-based Givens-weight representation for the lower triangular part and a column-based Givens-weight representation for the upper triangular part of the given rank structured matrix.

By induction, let us assume then that zeros have been created already in the first $k-1$ columns and rows of the matrix, $1 \leq k \leq n-1$. We assume that the Givens-weight representation of the lower triangular part is in weakly zero-creating form. We can then peel off its tails $G_{n-1,n}, \ldots, G_{k,k+1}$ to bring the $k$th column of the matrix in upper triangular form. During this process, the rank structure in the [illegible] triangular part can be updated in much the same way as in section 3, but now with the role of rows and columns interchanged. Having done this, we bring the Givens-weight representation of the upper triangular part back to its weakly zero-creating form. We can then peel off its tails $\tilde{G}^H_{n-1,n}, \ldots, \tilde{G}^H_{k+1,k+2}$ and apply them to the [illegible], to create zeros in the $k$th [illegible] of the matrix. During this process, the rank structure in the lower triangular part can be updated in the same way as in section 3, and so on.

Summarizing, we have now described how the Hessenberg reduction algorithm could be modified to a bidiagonal reduction algorithm. We now turn to a final modification of the algorithm.

**4.3. The case of nonpure structure blocks.** We assumed in section 3 that the structure blocks were [illegible] and lying strictly below the main diagonal. In this subsection, we will explain how the Hessenberg reduction algorithm can be adapted to work for nonpure structure blocks as well. Along this line, further in this subsection we will describe how to reduce a matrix to lower semiseparable plus diagonal instead of Hessenberg form.

First, we have to explain what the Givens-weight representation for such a nonpure rank structure looks like. We have to assume that the shift elements are [illegible] in the sense that two structure blocks intersecting each other on the main diagonal have the same shift element. If the shift elements are compatible, then we can just represent the rank structured matrix as $D_\lambda + A_{\text{pure}}$, where $D_\lambda$ is the diagonal matrix containing the shift elements, and $A_{\text{pure}}$ is a matrix satisfying a pure rank structure. The latter matrix can now be represented by a usual Givens-weight representation.

Let us explain how the nonpure Givens-weight representation, as just described, can be updated during the algorithm. For simplicity, we will restrict this explanation to the case where only the lower triangular part of the matrix is rank structured.

Starting from the example of Figure 4.5, the algorithm is shown in Figure 4.6.

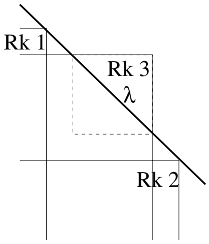


FIG. 4.5. *Starting nonpure rank structure.*

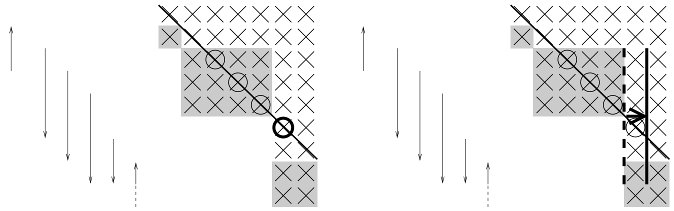
Let us comment on this figure. The starting situation is shown in Figure 4.6(a); it is assumed here that the tails of the subsequent Givens arrows until $G_{6,7}$ have already been peeled off from the representation. Hence the next operations are at the point of entering the nonpure diagonal part of the nonpure structure block $\mathcal{B}$.

Still concerning this figure, let us note that the circle notation expresses that, after spreading out the matrix, the value $\lambda$ should be added to each of the three originally circled entries of the structure block.

Since we are going to enter the nonpure part of $\mathcal{B}$, it is now the right moment to subtract the shift element $\lambda$ from the entry in the bottom right position of the structure block; this is indicated by the highlighted circle in Figure 4.6(a). Note that this subtraction will allow us to split a multiple of the identity matrix $\lambda I_4$ from the structure.

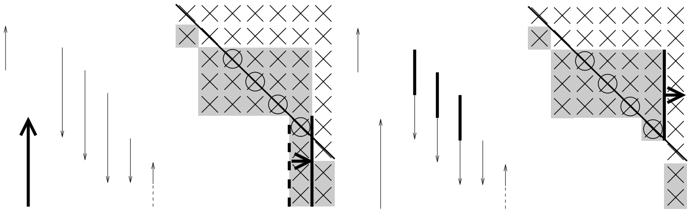
We will now apply the next operations. First, we enlarge the current action radius, and we apply an auxiliary unitary operation to bring the weights upwards; see Figures 4.6(b) and 4.6(c). Note that these operations can be performed exactly as in the shift-free case, since they involve only the [illegible] part of the representation, so that the shift element $\lambda$ is not involved.

We can then peel off the tails of the next Givens arrows and apply them to rows and columns. Since the shift element $\lambda$ appears only in the form of a multiple of the identity matrix $\lambda I_4$, which will clearly be preserved by this unitary similarity



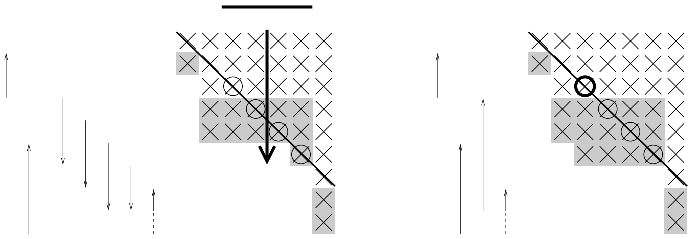
(a) Starting situation; subtract the shift element $\lambda$ from the indicated entry.

(b) Enlarge the action radius of the representation.



(c) Apply auxiliary unitary operations to bring the weights upwards.

(d) Apply the tails of the next Givens arrows to the rows.



(e) Multiply the transposed operations of Figure 4.6(b) to the columns.

(f) Add the shift element $\lambda$ to the indicated entry.

FIG. 4.6. *The case of nonpure structure blocks.*

operation, we can carry out these operations exactly as in the shift-free case; see Figures 4.6(d) and 4.6(e).

These peeling-off operations have caused the structure block $\mathcal{B}$ to move one position to the bottom right matrix corner. This means that the element which was originally situated in the top left position of $\mathcal{B}$ has now come for free. We can restore this element to its real-size form by adding the shift element $\lambda$ to it; see the highlighted circle in Figure 4.6(f). The next operations are not shown anymore.

Summarizing, we have now completely propagated through the nonpure structure block $\mathcal{B}$. During this process, the entry on the bottom right of the structure block was "brought into" the influence of the shift element in Figure 4.6(a), while the top left entry was “released” in Figure 4.6(f), corresponding to the structure block propagating to the bottom right matrix corner.

We note that similar techniques for manipulating shift elements under the action of unitary similarity operations were used in [17], for the special case of reducing an arbitrary matrix into a diagonal plus semiseparable matrix of semiseparability rank one.

The connection with the paper [17] can be made even tighter. More precisely, we will describe how the Hessenberg reduction algorithm can be modified so as to transform a given rank structured matrix $A$ into [illegible] form. Although this reduction could be achieved by first performing the Hessenberg reduction, and subsequently using this as input for the algorithm in [17], we will describe here an intermingled version of the algorithm.

For definiteness, let us denote with $\lambda_k \in \mathbb{C}$, $k = 1, \ldots, n$, the subsequent shift elements of the required, lower semiseparable plus diagonal structure. The corresponding structure blocks are defined as

$$\mathcal{B}_k : (i_k, j_k, r_k, \lambda_k) = (k, k, 1, \lambda_k).$$

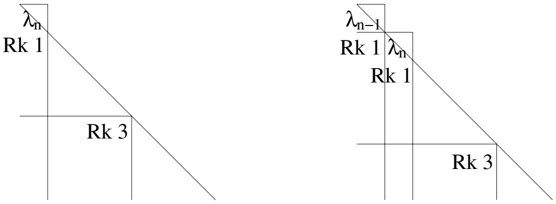
The algorithm proceeds by introducing the shift elements in the [illegible] order $\lambda_n, \ldots, \lambda_1$ on top of the matrix and chasing them to the bottom right. We can start this process by adding the [illegible] shift element $\lambda_n$ on top of the matrix, where it trivially satisfies a structure block $\mathcal{B} : (i, j, r, \lambda) = (1, 1, 1, \lambda_n)$; see Figure 4.7(a).

Let us note that in terms of the Givens-weight representation, this extra structure block $\mathcal{B}$ of Figure 4.7(a) can be incorporated in the factorization $D_\lambda + A_{\text{pure}}$ by placing the value $\lambda_n$ at the $(1, 1)$ element of $D_\lambda$ and subtracting it from the corresponding element of $A_{\text{pure}}$. It also implies an extra unitary transformation $U_0$ to be added to the Givens-weight representation of $A_{\text{pure}}$, in order to compress the [illegible] variant of this extra structure block, which we denote as $\mathcal{B}_{\text{pure}}$, by bringing its weight completely to the top row of the matrix $A_{\text{pure}}$. Since the structure block $\mathcal{B}_{\text{pure}}$ is Rk 1, the Givens arrows of $U_0$ will be of width one.

Now we create zeros in the first column of $A_{\text{pure}}$ by applying the tails of the subsequent Givens arrows $G_{n-1,n}, \ldots, G_{1,2}$ to the rows and their transposes $G^H_{n-1,n}, \ldots, G^H_{1,2}$ to the columns. Since the first column is also involved in this process, the created zeros in the first column will be destroyed again. On the other hand, each of the existing structure blocks will be chased one position to the bottom right position, according to the implementation in Figure 4.6. In particular, it follows that the extra structure block $\mathcal{B}$ will transform into a new structure block $\widetilde{\mathcal{B}} : (i, j, r, \lambda) = (2, 2, 1, \lambda_n)$; see Figure 4.7(b).

Suppose then that we have already created $k-1$ lower semiseparable plus diagonal structure blocks in the top left corner of the matrix, and suppose that we are at the point of applying the $k$th upward sweep of Givens transformations, $2 \leq k \leq n-1$. Before doing this, we add the new shift element $\lambda_{n+1-k}$ on top of the matrix, where it trivially satisfies a structure block $\mathcal{B} : (i, j, r, \lambda) = (1, 1, 1, \lambda_{n+1-k})$. We can then correspondingly update the Givens-weight representation. Next, we can do the same as above, i.e., we apply similarity transformations based on the subsequent tails of the Givens arrows $G_{n-1,n}, \ldots, G_{1,2}$ of the matrix $A_{\text{pure}}$ in order to chase each of the existing structure blocks one position to the bottom right matrix corner.

Note that at the end of this process, the given rank structured matrix will be brought completely into lower semiseparable plus diagonal form, with the required shift elements $\lambda_k$.



(a) (b)

Fig. 4.7. *The left picture shows the starting situation for the lower semiseparable plus diagonal reduction algorithm. The first extra structure block $\mathcal{B}$ has already been added to the top left corner. The right picture shows the situation after applying the first sweep of Givens transformations of the lower semiseparable plus diagonal reduction algorithm. Note that the structure blocks have moved to the bottom right corner, and that the next extra structure block has already been added to the top left corner.*

Let us point out one more thing about the previous reduction algorithm. In principle, one should be cautious about the validity of the structure propagation mechanism for the trivially satisfied structure block $\mathcal{B}$ in the top left matrix corner, since this structure block involves the first row of the matrix, so that the result of Theorem 11 does [illegible] hold anymore. Still, the structure propagation argument [illegible] remain valid, by the fact that, by construction, the applied Givens transformations created zeros in the first column of the [illegible] component $A_{\text{pure}}$, rather than $A$ itself, and this was precisely the necessary condition needed in the first paragraph of the proof of Theorem 11.

[illegible] 13.

1. The presence of rank structure in the upper triangular part was left out of the previous discussion, but can be handled similarly as before. Let us assume, e.g., that the matrix is Hermitian. Then we will suffice with having access over the Givens-weight representation of the lower triangular part, having now a "bulge" at each place where the structure goes beyond the main diagonal. Let us consider, e.g., the situation in Figure 4.6(b). Here we need to have access over the three top elements lying in between the two thick vertical lines. Each of these elements could then be computed by the Hermitian character, using the knowledge of the corresponding element in the lower triangular part, which can be obtained by means of some auxiliary spreading-out operations of the corresponding column of the lower triangular part (one should use an auxiliary variable for doing this). Let us note that these spreading-out operations become more expensive when the required element is situated further from the main diagonal.
2. The Hessenberg reduction algorithm can be seen as a special case of the lower semiseparable plus diagonal reduction algorithm, provided that one chooses $\lambda_k = \infty$ for each $k$, in the sense described in [5].
3. When the required diagonal plus semiseparable structure is close to Hessenberg form, i.e., $\lambda_k \approx \infty$, it can be expected that numerical problems may arise due to large values of the $\lambda_k$. To solve this problem, we may observe that the only place where information about the shift elements is needed is for the determination of each top Givens transformation $G_{1,2}$. Once this has been

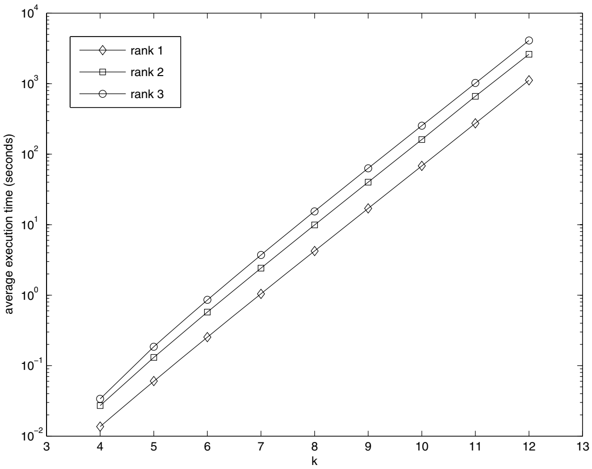


FIG. 5.1. *Average execution time for five random samples of size* $n = 2^k$ *and rank* $r = 1, 2, 3$.

done, the application of this and other Givens transformations can be applied in a "shift-free" way, by working with the induced pure structure blocks, lying just below the main diagonal. Our numerical experiments indicate that this shift-free variant is indeed more stable than the variant described previously, even for moderate sizes of the shift elements $\lambda_k$.

**5. Numerical experiments.** In this section we report on the results of some numerical experiments. The algorithms were implemented in MATLAB.[1] The experiments were executed on an Intel PC running MATLAB Version 7.0.1.24704 (R14) under Linux having 1GByte of memory and an Intel Pentium 4 processor running at 3.2 GHz. The software of these experiments can be requested from the authors.

We constructed symmetric rank structured matrices in $\mathbb{R}^{n \times n}$, with $n = 2^k$ for $k = 4, \ldots, 12$. Starting from a diagonal matrix containing the desired eigenvalues, which were uniformly randomly chosen in the interval $[-1, 1]$, we applied to this matrix a similarity transformation based on a "disturbing" sequence of Givens arrows of width $r$. This resulted in a rank structure whose structure blocks are situated just below the main diagonal, following immediately one after the other. The upper triangular part is then known by symmetry.

Since it can be argued that the above construction yields rather special rank structured matrices, we next applied a "randomization" procedure. We did this by applying an additional similarity transformation based on Givens transformations, computed in a structure-preserving way. Note that symmetry is preserved during all these operations. A detailed description of this randomization method will not be given here.

For each size $n$, the above scheme was carried out for subsequent rank indices $r = 1, 2, 3$. For each of these sizes and each of these rank indices, there were considered five samples. Figure 5.1 shows for each size $n = 2^k$ and each rank index $r$ the execution time $T_{k,r}$ averaged over the five samples of performing the reduction to Hessenberg form.

To check that the computational complexity is quadratic in the size of the matrix,

---

[1] MATLAB is a registered trademark of The MathWorks, Inc.

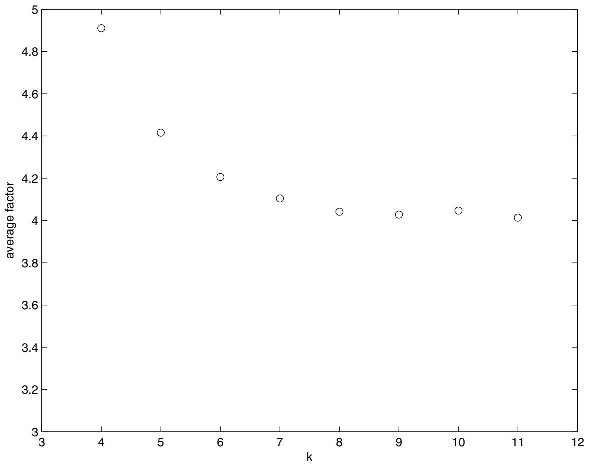


FIG. 5.2. *Fraction* $T_{k+1,r}/T_{k,r}$ *averaged over five random samples and over ranks* $r = 1, 2, 3$ *in function of the size* $n = 2^k$.

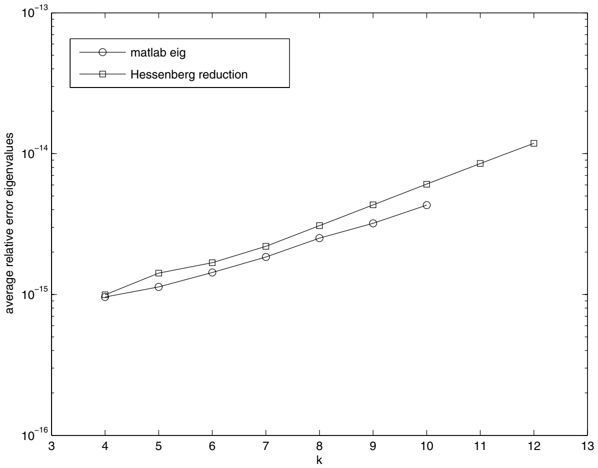


FIG. 5.3. *Relative* 2-norm of the error of the eigenvalues averaged over five random samples and over ranks $r = 1, 2, 3$ in function of the size $n = 2^k$.

Figure 5.2 shows the fraction $T_{k+1,r}/T_{k,r}$ averaged over the five samples and over the ranks $r = 1, 2, 3$.

To measure the accuracy of the algorithm, we computed the relative norm $\frac{||\lambda-\lambda_0||_2}{||\lambda_0||_2}$, where $\lambda_0, \lambda \in \mathbb{R}^n$ denote the vectors containing the exact and computed eigenvalues. Figure 5.3 shows the relative 2-norm of the error averaged over the five samples and the rank indices $r = 1, 2, 3$.[2]

The computation of Givens transformations during the algorithm was performed according to the implementation described in [9, section 5.1]. We note that the accuracy of the algorithm turns out to be slightly sensitive to the choice of the used

---

[2] The eigenvalues of the resulting tridiagonal matrix at the end of the Hessenberg reduction process were computed using the MATLAB routine "trideig" due to P.-O. Persson. This routine is available at http://www.mit.edu/~persson/mltrid/index.html.

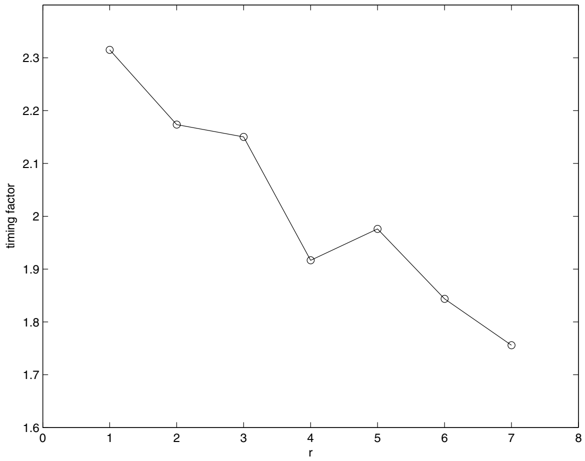


FIG. 5.4. *Fraction $T_{2r}/T_r$ for size $n = 2^9$ in function of the rank index $r = 2^l$ with $l = 0, 1, \dots, 6$.*

Givens routine, but this effect is rather modest.

The pull-through lemma was implemented in the trivial way. This means that, using the notations of Lemma 7, we explicitly computed the 3 by 3 matrix $G'_{1,2}G_{2,3}G_{1,2}$ in its full form. Subsequently, we computed Givens transformations such that

$$(\tilde{G}_{2,3})^H(\tilde{G}_{1,2})^H(\tilde{G}'_{2,3})^H G'_{1,2}G_{2,3}G_{1,2} \tag{5.1}$$

is upper triangular, with positive diagonal elements. Hence (5.1) is a unitary, upper triangular matrix with positive diagonal elements, so that this must be the identity matrix. Rewriting this fact leads to the desired refactorization of Lemma 7. We note that our future work will describe in more detail the implementation of the pull-through lemma, including a nontrivial speed-up which was already implicit in [10].

To check that the computational complexity is linear as a function of the rank index $r$, we considered the execution time $T_r$ for matrices of fixed size $n = 2^9 = 512$ and varying rank index $r = 2^l$ with $l = 0, 1, \dots, 7$. Figure 5.4 gives the fraction $T_{2r}/T_r$ for subsequent rank indices. Note that the fraction tends to approximate 2 for large rank indices $r$. In fact, the fraction tends to go even below this value of 2; but we note that this is artificial since for high ranks the ranks become relatively large with respect to the size $n = 2^9$, so that the distribution of Givens transformations on the "borders" of the matrix start to have a nonneglible impact on the timings.

We note that similar experiments have also been performed for other kinds of rank structures, which are less regular than the one we took for the above numerical experiments. The results of these experiments were similar to those reported above.

**6. Conclusion.** In this paper we described an algorithm for performing the Hessenberg reduction of rank structured matrices. Numerical experiments indicated that this approach leads to a stable computation of the eigenvalues of the given matrix. We showed that the algorithm has complexity $O((ar + bs)n^2)$ in case of a Hermitian plus low rank matrices for suitable $a, b \in \mathbb{R}$. We explained that this complexity holds also for unitary plus low rank matrices, provided that one agrees to perform numerical approximations during the algorithm. Our future work includes an alternative algorithm for the Hessenberg reduction of unitary and related matrices, using an appropriate representation.

## REFERENCES

[1] S. Chandrasekaran and M. Gu, *Fast and stable eigendecomposition of symmetric banded plus semi-separable matrices*, Linear Algebra Appl., 313 (2000), pp. 107–114.

[2] S. Delvaux and M. Van Barel, *Structures preserved by the QR-algorithm*, J. Comput. Appl. Math., 187 (2005), pp. 29–40.

[3] S. Delvaux and M. Van Barel, *A Givens-Weight Representation for Rank Structured Matrices*, SIAM J. Matrix Anal. Appl., to appear.

[4] S. Delvaux and M. Van Barel, *Rank structures preserved by the QR-algorithm: The singular case*, J. Comput. Appl. Math., 189 (2006), pp. 157–178.

[5] S. Delvaux and M. Van Barel, *Structures preserved by matrix inversion*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 213–228.

[6] P. Dewilde and A.-J. van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.

[7] D. Fasino, N. Mastronardi, and M. Van Barel, *Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form*, Contemp. Math., 323 (2003), pp. 105–118.

[8] G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[10] W. B. Gragg, *The QR algorithm for unitary Hessenberg matrices*, J. Comput. Appl. Math., 16 (1986), pp. 1–8.

[11] W. Hackbusch, B. N. Khoromskij, and S. A. Sauter, *On $\mathcal{H}^2$-matrices*, in Lect. Appl. Math., H. Bungartz and L. Horsten, eds., Springer-Verlag, Berlin, 2000, pp. 9–29.

[12] P. G. Martinsson, V. Rokhlin, and M. Tygert, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl., 50 (2005), pp. 741–752.

[13] N. Mastronardi, S. Chandrasekaran, and S. Van Huffel, *Fast and stable algorithms for reducing diagonal plus semiseparable matrices to tridiagonal and bidiagonal form*, BIT, 41 (2001), pp. 149–157.

[14] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics Appl. Math. 20, SIAM, Philadelphia, 1998.

[15] H. Rutishauser, *On Jacobi rotation patterns*, in Proceedings of Symposia in Applied Mathematics, Experimental Arithmetic, High Speed Computing, and Mathematics 15, AMS, Providence, RI, 1963, pp. 219–239.

[16] H. R. Schwartz, *Tridiagonalization of a symmetric band matrix*, Numer. Math., 12 (1968), pp. 231–241.

[17] R. Vandebril, E. Van Camp, M. Van Barel, and N. Mastronardi, *Orthogonal similarity transformation of a symmetric matrix into a diagonal-plus-semiseparable one with free choice of the diagonal*, Numer. Math., 102 (2006), pp. 709–726.

[18] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, UK, 1965.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 927–953

# BLOCK AND PARALLEL VERSIONS OF ONE-SIDED BIDIAGONALIZATION*

NELA BOSNER† AND JESSE L. BARLOW‡

**Abstract.** Two new algorithms for one-sided bidiagonalization are presented. The first is a block version which improves execution time by improving cache utilization from the use of BLAS 2.5 operations and more BLAS 3 operations. The second is adapted to parallel computation. When incorporated into singular value decomposition software, the second algorithm is faster than the corresponding ScaLAPACK routine in most cases. An error analysis is presented for the first algorithm. Numerical results and timings are presented for both algorithms.

**Key words.** singular value decomposition, bidiagonalization, block algorithm, parallel algorithm, numerical analysis

**AMS subject classifications.** 15A18, 65F30, 68W10

**DOI.** 10.1137/050636723

## 1. Introduction.

There are two main types of algorithms for computing the complete singular value decomposition (SVD) of a matrix $A$: one-sided Jacobi methods [12] and algorithms based upon bidiagonalization. Recently there have been significant improvements in both types of methods [2], [3], [17], [18], [23], [24]; this work considers bidiagonalization-based algorithms. Such algorithms use orthogonal transformations to obtain a bidiagonal form and then apply a fast algorithm to obtain the SVD of the bidiagonal matrix [19], [24], [32].

Let $A \in \mathbb{R}^{m\times n}$, $m \geq n$ (for $m < n$, consider $A^T$). Then there exist a [illegible] matrix $U \in \mathbb{R}^{m\times n}$ (i.e., $U^TU = I_n$), an [illegible] matrix $V \in \mathbb{R}^{n\times n}$ (i.e., $V^TV = VV^T = I_n$), and a bidiagonal matrix

$$B = \begin{bmatrix} \psi_1 & \phi_2 & & & \\ & \psi_2 & \phi_3 & & \\ & & \ddots & \ddots & \\ & & & \psi_{n-1} & \phi_n \\ & & & & \psi_n \end{bmatrix} \tag{1.1}$$

such that

$$A = UBV^T. \tag{1.2}$$

Equation (1.2) describes [illegible] of the matrix $A$. There are several algorithms for bidiagonalization of a matrix, and they differ in the way they construct the matrices $U$ and $V$. The most commonly used algorithms for computing (1.2) are given by Golub and Kahan in [20], and Golub and Reinsch in [21], where the matrices $U$ and $V$ are products of Householder reflectors. Lawson and Hanson in [27,

---

*Received by the editors July 23, 2005; accepted for publication (in revised form) by B. J. Kågström May 10, 2007; published electronically August 31, 2007.
http://www.siam.org/journals/simax/29-3/63672.html

†Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia (nela@math.hr). This author's research was supported by Croatian MZOS grant 0037120.

‡Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802-6822 (barlow@cse.psu.edu). This author's research was supported by NSF grant CCF-0429481.

p. 119] and Chan in [10] present a modification of the Golub–Kahan algorithm, which performs a QR factorization of the matrix $A$ before bidiagonalization, resulting in an algorithm with fewer operations if $m > 5/3n$ and $U$ is not explicitly formed. A bidiagonalization algorithm based on Givens rotations, which can attain higher accuracy, is given by Barlow [2]. Grosser and Lang in [23] propose a two-stage method, where in the first stage a matrix is reduced to a banded matrix and then in the second stage a banded matrix is bidiagonalized. This algorithm performs the majority of calculations in matrix-matrix products and can be faster than the ScaLAPACK bidiagonalization routine. Another different approach is described in [20], where the bidiagonal form is obtained by means of the Lanczos algorithm. This approach is fast but can be numerically unstable. Still there are many applications that do not suffer from this instability, and there is a simple procedure that corrects the possible loss of orthogonality; cf. [31].

Recently, in [28], [30], and [29] Ralha proposed a new approach for bidiagonalizing a matrix, the so-called [illegible]. The main idea is to implicitly tridiagonalize the matrix $A^TA$:

$$V^TA^TAV = T, \qquad T = \text{tridiagonal matrix},$$

by a one-sided orthogonal transformation of $A$

$$F = AV.$$

$V$ is computed as a product of Householder reflectors without explicitly forming $A^TA$. Then the Gram–Schmidt QR factorization of the matrix $F$ is performed to obtain

$$UB = F = AV,$$

where $B$ is upper triangular. In the case when $A$ has full rank, the matrix $B$ is the Cholesky factor of $T$, that is, $T = B^TB$, and hence $B$ must be bidiagonal. That property results in a fast bidiagonalization algorithm, which is much more suitable for parallel computing than the standard bidiagonalization algorithms. Here the modified Gram–Schmidt QR factorization of $F$ is computed by orthogonalizing the $k$th column of $F$ against the $(k-1)$th column of $U$ and normalizing. Unfortunately, this procedure is not always numerically stable and may lead to a matrix $U$ that is far from being left orthogonal in finite precision arithmetic.

To improve Ralha's algorithm, Barlow, Bosner, and Drmač [3] proposed a modification, where one step of Gram–Schmidt orthogonalization and postmultiplication with one Householder reflector are performed simultaneously, resulting in a direct bidiagonalization algorithm. This algorithm produces exactly the same result as Ralha's algorithm in exact arithmetic, but it turns out to be numerically stable in finite precision arithmetic.

The rest of this paper is organized as follows. The recently developed one-sided bidiagonalization from [3] is presented in section 2, together with a bound on its backward error. In section 3, a block version of the new one-sided bidiagonalization is introduced, and a detailed numerical analysis is given in Theorem 3.4. The results of numerical tests regarding efficiency of the block one-sided bidiagonalization are presented in section 4. Sections 5 and 6 deal with parallel versions of the new one-sided bidiagonalization and its efficiency.

## 2. The new stable one-sided bidiagonalization.

The main difference between Ralha's algorithm and the new stable algorithm proposed in [3] is that, in the new bidiagonalization algorithm, transformations with Householder reflectors and the Gram–Schmidt orthogonalization are interlaced, whereas Ralha's bidiagonalization separates these processes. The criteria for choosing Householder reflectors are also different.

Let $u_k$ be the $k$th column of the matrix $U$, and let $U_k = [u_1, \ldots, u_k]$ be a matrix containing the first $k$ columns of $U$. Further, let $V_k$ denote a Householder reflector such that

$$V_k = I - v_k v_k^T, \qquad \text{where } \|v_k\|_2 = \sqrt{2}.$$

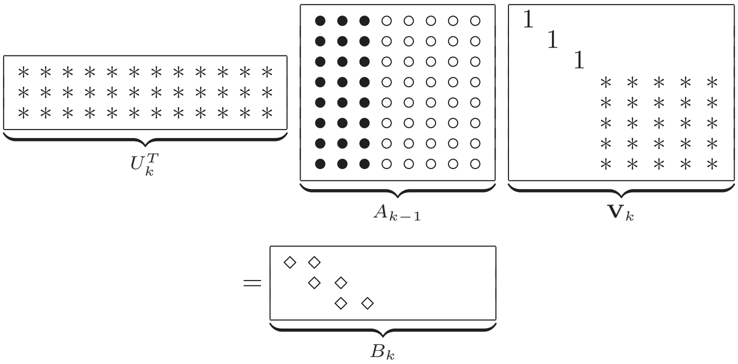
Then the new stable bidiagonalization can be described in its simplest form as follows:

- $A_0 = A$.
- For $k = 1, 2, \ldots,$
  - ○ $u_k$ is produced from the $k$th column of $A_{k-1}$ by orthogonalization against $u_{k-1}$ (if $k > 1$) and normalization

    $$U_k = [u_1, \ldots, u_k], \quad k = 1, \ldots, n.$$

  - ○ Householder reflector $\mathbf{V}_k$ is chosen so that

    $$U_k^T A_{k-1} \mathbf{V}_k = B_k \in \mathbb{R}^{k \times n}, \qquad B_k \text{ is bidiagonal,} \tag{2.1}$$



    and the matrix $A_{k-1}$ is postmultiplied with $\mathbf{V}_k$

    $$A_k = A_{k-1} \mathbf{V}_k, \quad k = 1, \ldots, n-2.$$

- End of loop.
- $V$ is produced by accumulation of the Householder reflectors

$$V = \mathbf{V}_1 \cdots \mathbf{V}_{n-2}, \qquad F = A_{n-2} = AV. \tag{2.2}$$

[illegible] 2.1. The elements of $A_{k-1}$ denoted by $\circ$ are used in the current step of the algorithm to compute the vector $z_k$, which determines the Householder reflector $\mathbf{V}_k$. The Householder reflector $\mathbf{V}_k$ is defined as

$$\mathbf{V}_k = \begin{bmatrix} I_k & 0 \\ 0 & V_k \end{bmatrix}, \tag{2.3}$$

and the Householder reflector $V_k \in \mathbb{R}^{(n-k)\times(n-k)}$ is chosen so that $V_k z_k = \pm\|z_k\|_2 e_1$. The elements denoted by $\bullet$ are computed columns of $F$, and in the next steps they remain unchanged. The computed elements of $B$ are denoted by $\diamond$.

[illegible] 2.2. In order to produce $U$ with $n$ columns, the matrix $A$ should have full column rank. If rank $(A) < n$, then $\psi_k = 0$ for some $k$. In [3], it is shown that this case can be easily handled by an $O(mn+n^2)$ postprocessing procedure that produces a decomposition

$$A = U_c B_c V_c^T,$$

where $U_c \in \mathbb{R}^{m\times p}$ and $V_c \in \mathbb{R}^{n\times p}$ are left orthogonal, $B_c \in \mathbb{R}^{p\times p}$ is upper bidiagonal and nonsingular, and $p = \mathrm{rank}(A)$.

The following pseudocode provides the details of the described algorithm.[1]

ALGORITHM 2.1. [illegible] $A \in \mathbb{R}^{m\times n}$ $\mathrm{rank}(A) = n > 2$ [illegible] $U = [u_1, \dots, u_n]$ [illegible] $B$ [illegible] (1.1) [illegible] $V = V^{(n-2)}$ [illegible] $A = UBV^T$

```
(1)  $A_0 = A$.
(2)  $f_1 = A(:,1)$.  $\psi_1 = \|f_1\|_2$.
(3)  $u_1 = f_1/\psi_1$.
for $k = 1: n-2$
    (4) $z_k = A_{k-1}(:,k+1:n)^T u_k$.
    (5) $[\phi_{k+1}, v_k] =$ householder$(z_k)$.
    (6) $A_k(:,1:k) = A_{k-1}(:,1:k)$.
    (7) $A_k(:,k+1:n) = A_{k-1}(:,k+1:n) - A_{k-1}(:,k+1:n)v_k v_k^T$.
    (8) $f_{k+1} = A_k(:,k+1)$.
    (9) $s_{k+1} = f_{k+1} - \phi_{k+1}u_k$.  $\psi_{k+1} = \|s_{k+1}\|_2$.
    (10) $u_{k+1} = s_{k+1}/\psi_{k+1}$.
end.
(11)  $f_n = A_{n-2}(:,n)$.  $\phi_n = u_{n-1}^T f_n$.
(12)  $s_n = f_n - \phi_n u_{n-1}$.  $\psi_n = \|s_n\|_2$.
(13)  $u_n = s_n/\psi_n$.
(14) $V^T =$ householder_product$(v_1, \dots, v_{n-2})$
end.
```

The auxiliary functions **householder**() and **householder_product**() are defined as follows.

**function** $[\phi, v]$=**householder**$(z)$
{[illegible] **householder**() [illegible] $\phi$ [illegible] $v$ [illegible] $\|v\|_2 = \sqrt{2}$ [illegible] [25, [illegible] 19]. The Householder reflector is then formed as $V = I - vv^T$ with property that $Vz = \phi e_1$.}

**function** $V^T =$ **householder_product**$(v_1, \dots, v_n)$
{[illegible] **householder_product**() [illegible] $V^T$ [illegible] $n$ [illegible] $V^T = \mathbf{V}_n \cdots \mathbf{V}_1$ $\mathbf{V}_k$ [illegible] $k = 1, \dots, n$ [illegible] (2.3) [illegible] $V_k = I - v_k v_k^T$ [illegible] `sorgbr()` [1] }

In [3], the following theorem about the numerical stability of Algorithm 2.1 was proved. Here we denote the unit roundoff with $\varepsilon$.

THEOREM 2.1. [illegible] $\tilde{B}$ [illegible] 2.1 [illegible] $(m+n)\times(m+n)$ [illegible] $\mathcal{P}$ [illegible] $n\times n$ [illegible] $V$

---

[1] FORTRAN routines for the SVD using the one-sided bidiagonalization methods described in this paper are available from the first author.

[illegible] $\Delta A$ $\delta A$ [illegible]

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{\mathcal{P}}^T \begin{bmatrix} \Delta A \\ A + \delta A \end{bmatrix} \hat{V}, \quad \left\| \begin{bmatrix} \Delta A \\ \delta A \end{bmatrix} \right\|_F \leq \xi \|A\|_F, \tag{2.4}$$

[illegible] $\xi = O(mn + n^3)\varepsilon$ [illegible] $\tilde{V}$ [illegible] $\hat{V}$ [illegible] $\|\tilde{V} - \hat{V}\|_F \leq O(n^2)\varepsilon$ [illegible] $\breve{U}$ [illegible] $\breve{\delta A}$ [illegible]

$$A + \breve{\delta A} = \breve{U} \tilde{B} \hat{V}^T, \quad \|\breve{\delta A}\|_F \leq \sqrt{2}\xi \|A\|_F. \tag{2.5}$$

Although this result implies that Algorithm 2.1 is numerically stable for computing $B$ and $V$, we cannot guarantee that the computed matrix $U$ is numerically orthogonal. In many circumstances, the possible loss of orthogonality is irrelevant [8]. One can also get nearly orthogonal bases for the parts associated with the leading singular values of $A$ [3, Theorem 3.19 and Corollary 3.20].

Another important characteristic of an algorithm is its efficiency, and the preferable way to evaluate efficiency is through execution time. The execution time of a numerical algorithm depends on two properties: the floating point operation count and the time spent on communication among hierarchically organized computer memory. We are concerned about the efficiency of full SVD algorithms that include bidiagonalization, when computing all of the SVD factors: $P$, $Q$, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ such that $A = P\Sigma Q^T$.

Extensive numerical tests were performed to test the efficiency of the SVD algorithms. Even though an SVD routine with Algorithm 2.1 requires fewer floating point operations (see [3]), a straightforward implementation of Algorithm 2.1 is often slower than the current LAPACK [1] routine in `sgesvd()`, since the LAPACK routine optimizes cache memory usage, whereas Algorithm 2.1 does not. In order to decrease cache communication time, we develop a block version of Algorithm 2.1 next.

**3. Block version of the new stable one-sided bidiagonalization.** The new block version of Algorithm 2.1 improves the usage of fast cache memory by performing as many operations as possible on the data that are currently stored in the cache. In order to do that, one has to transform the original algorithm. The first modification of the algorithm is that transformations by Householder reflectors are aggregated, where the WY representation is used for a product of Householder reflectors [4]. In the WY representation the product of $j$ Householder reflectors is presented as $V_j \cdots V_1 = I - WY^T$, where $W$ and $Y$ have $j$ columns. This means that the matrix $A$ is updated after every $b$ steps, where $m \times b$ is the block dimension. Most of the operations in Algorithm 2.1 are matrix-vector operations, coded as BLAS 2 operations [13]. Memory hierarchy is utilized more efficiently if such algorithms are written in terms of matrix-matrix operations, coded as BLAS 3 operations [14], [15], or grouped matrix-vector operations, called BLAS 2.5 operations [11], [26]. Employing the WY representation of products of Householder transformations results in more BLAS 3 operations; using the BLAS 2.5 approach of Howell et al. [26] leads to further improvement. Operations on the same data, but performed in different places in Algorithm 2.1, are now performed simultaneously. These operations are

$$\boxed{\begin{array}{rcl} x & \leftarrow & x + A^T y \\ w & \leftarrow & Ax \end{array}} \quad \text{or} \quad \boxed{\begin{array}{rcl} A & \leftarrow & A + uv^T \\ x & \leftarrow & A^T y \\ w & \leftarrow & Ax \end{array}}\,. \tag{3.1}$$

Now we discuss our modifications of Algorithm 2.1. As an input to the algorithm we will take the matrix $A \in \mathbb{R}^{m\times n}$ and partition it into block columns. Let $n = b\cdot g+r$, $r \leq b+1$, where $b$ is a given block column dimension and $g = \lfloor (n-2)/b \rfloor$ is the number of blocks of dimension $m\times b$. We choose the last two columns to be outside of the block partition, because the last two steps of the one-sided bidiagonalization (corresponding to the last two columns) do not involve computation of a Householder reflector. The $g$ blocks will be updated by means of aggregated Householder transformations and BLAS 2.5 transformations related to the first group of transformations in (3.1). The remaining $r = n - b \cdot g$ columns will be updated with nonaggregated Householder transformations and the second group of BLAS 2.5 transformations in (3.1). As each block consists of $b$ columns, the steps of the algorithm will be organized in two loops: the outer loop going through $g$ blocks and the inner loop going through $b$ columns of the block. Thus we will denote by $A_{j,k}$ the matrix $A$ after the first $j-1$ blocks, and the first $k$ columns in the $j$th block have been updated.

The main difference between the stable bidiagonalization and its block version is the way Householder reflectors are computed and applied to the matrix $A$. In the $k$th step of Algorithm 2.1, columns $k+1$ through $n$ of the matrix $A$ are updated with the Householder reflector $\mathbf{V}_k$. After this step, the $(k+1)$th column is not changed anymore and is consequently equal to the $(k+1)$th column of the matrix $F$ defined in (2.2).

In the block version of the stable bidiagonalization, updates with Householder reflectors are done blockwise. This means that, only when all of the columns in one block are updated and assigned to $F$ (they will not be modified in the next steps), the rest of the matrix will then be updated with $b$ Householder reflectors in aggregated form that correspond to $b$ steps of Algorithm 2.1. Until then, only the current column is updated. Let us assume that we have computed the first $(j-1)$ blocks of the matrix $F$ obtaining the matrix $A_{j,0}$ and that we are observing the operations in the $j$th step of the outer loop. Then for $k = 2, \ldots, b$ only the $((j-1)b+k)$th column is updated by Householder reflectors from the steps $1, \ldots, k-1$ of the same block, obtaining the matrix $A_{j,k}$, and a new Householder reflector $\mathbf{V}_{(j-1)b+k}$ is computed. $\mathbf{V}_{(j-1)b+k}$ will affect the columns $((j-1)b+k+1)$ through $n$, but no updates are done. The matrix $A_{j,1}$ is equal to $A_{j,0}$ because the $((j-1)b+1)$th column is already updated; only the Householder reflector $\mathbf{V}_{(j-1)b+1}$ is computed. We use the WY form for a product of Householder reflectors described in [4] to write

$$\mathbf{V}_{(j-1)b+1} \ldots \mathbf{V}_{(j-1)b+k-1} = I - Y_j(:, 1\colon k-1) W_j(:, 1\colon k-1)^T. \tag{3.2}$$

After the $(jb)$th column has been updated, the columns $jb+1$ through $n$ are updated with the product $\mathbf{V}_{(j-1)b+1}, \ldots, \mathbf{V}_{jb}$ in WY form (3.2). This process is illustrated in Figure 3.1 with notation defined in (3.3). The $(g+1)$th block is updated with Householder reflectors in the usual way, as it is done in Algorithm 2.1.

This is the same approach as in the LAPACK routine `sgebrd()` [16], where the routine `slabrd()` is called first, followed by the routine `sgebd2()`. `slabrd()` performs the two-sided aggregated Householder transformation over the first $g$ blocks, and `sgebd2()` performs the unblocked transformations. The differences are that in the block version of Algorithm 2.1 only one-sided Householder transformations are performed and that the dimension of the block is computed differently.

Aggregated Householder transformations represent only one modification of Algorithm 2.1. The other modification is achieved by using the ideas described in [26].

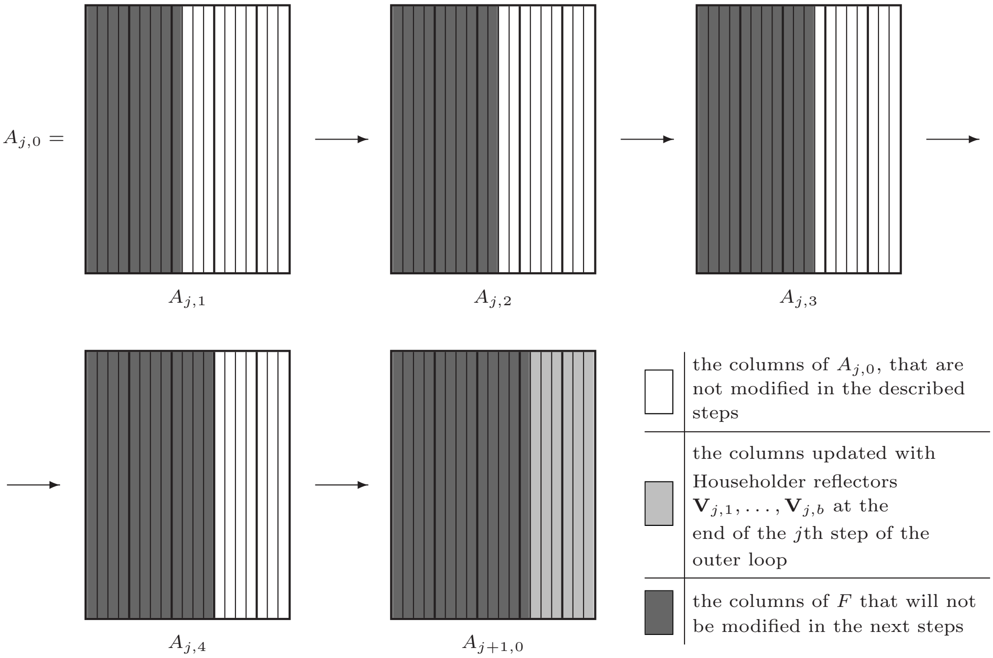

$A_{j,0} =$
$A_{j,1}$
$A_{j,2}$
$A_{j,3}$
$A_{j,4}$
$A_{j+1,0}$
the columns of $A_{j,0}$, that are not modified in the described steps
the columns updated with Householder reflectors $\mathbf{V}_{j,1}, \ldots, \mathbf{V}_{j,b}$ at the end of the $j$th step of the outer loop
the columns of $F$ that will not be modified in the next steps

FIG. 3.1. *Column update in the $j$th block of the matrix $A$.*

Let us define the following correspondence:

$$\begin{aligned} &\ell = (j-1)b + k, && \text{current } \ell\text{th column is the } k\text{th column in the } j\text{th block,} \\ &\ell \leftrightarrow (j,k) && \text{the indices with } \ell \text{ are replaced by } (j,k). \end{aligned} \tag{3.3}$$

This correspondence is introduced only for notational convenience. Now we will investigate the lines **(4)**, **(5)**, and **(7)** in Algorithm 2.1, but with the index $k$ replaced by $\ell$. In all of these statements the vector $z_\ell \to z_{j,k}$ is directly or indirectly used. In line **(4)** $u_\ell$ is multiplied by $A_{\ell-1}(:, \ell+1\colon n)^T \to A_{j,k-1}(:, \ell+1\colon n)^T$ in order to obtain $z_{j,k}$. On the other hand, in line **(7)** the vector $v_\ell \to v_{j,k}$ is multiplied by $A_{j,k-1}(:, \ell+1\colon n)$, and $v_{j,k}$ is realized from $z_{j,k}$ through line **(5)** and the function **householder**(). From the definition of **householder**(), we have

$$\begin{aligned} z_{j,k} &= A_{j,k-1}(:, \ell+1, n)^T u_\ell, \\ v_{j,k} &= \frac{\sqrt{2}(z_{j,k} - \phi_{\ell+1}e_1)}{\|z_{j,k} - \phi_{\ell+1}e_1\|_2}, \quad \text{and thus} \end{aligned} \tag{3.4}$$

$$A_{j,k-1}(:, \ell+1\colon n)v_{j,k} = \frac{\sqrt{2}[A_{j,k-1}(:, \ell+1\colon n)z_{j,k} - \phi_{\ell+1}A_{j,k-1}(:, \ell+1)]}{\|z_{j,k} - \phi_{\ell+1}e_1\|_2}.$$

From the previous observations concerning the update of the matrix $A_{j,0}$ with Householder reflectors, in the $\ell$th step (which in the block version will correspond to the $j$th step of the outer loop and the $k$th step of the inner loop) columns $\ell+1, \ldots, n$ are not yet updated. Since $A_{j,k-1} = A_{j,0}\mathbf{V}_{j,1} \ldots \mathbf{V}_{j,k-1}$, (3.2) and (3.4) imply that

$$\begin{aligned} z_{j,k} = {} & A_{j,0}(:, \ell+1, n)^T u_\ell \\ & -W_j(\ell+1\colon n, 1\colon k-1)Y_j(:, 1\colon k-1)^T A_{j,0}^T u_\ell, \end{aligned} \tag{3.5}$$

$$\begin{aligned} A_{j,k-1}(:\,,\ell+1\colon n)v_{j,k} &= A_{j,0}(:\,,\ell+1\colon n)v_{j,k} \\ &\quad -A_{j,0}Y_j(:\,,1\colon k-1)W_j(\ell+1\colon n,1\colon k-1)^T v_{j,k} \\ &= \frac{\sqrt{2}[A_{j,0}(:\,,\ell+1\colon n)z_{j,k}-\phi_{\ell+1}A_{j,0}(:\,,\ell+1)]}{\|z_{j,k}-\phi_{\ell+1}e_1\|_2} \\ &\quad -A_{j,0}Y_j(:\,,1\colon k-1)W_j(\ell+1\colon n,1\colon k-1)^T v_{j,k}. \end{aligned} \tag{3.6}$$

If we define

$$z_{j,k}^{(1)} = -W_j(\ell+1\colon n,1\colon k-1)Y_j(:\,,1\colon k-1)^T A_{j,0}^T u_\ell$$

as the first phase in the computation of $z_{j,k}$ and

$$x_{j,k}^{(1)} = A_{j,0}(:\,,\ell+1\colon n)z_{j,k}$$

as the first phase in the computation of the vector $x_{j,k}^{(4)} = A_{j,k-1}(:\,,\ell+1\colon n)v_{j,k}$, then

$$\boxed{\begin{array}{rcl} z_{j,k} & = & z_{j,k}^{(1)} + A_{j,0}(:\,,\ell+1\colon n)^T u_\ell \\ x_{j,k}^{(1)} & = & A_{j,0}(:\,,\ell+1\colon n)z_{j,k} \end{array}} \qquad \begin{array}{l} \text{from (3.5)} \\ \text{from (3.6)} \end{array}$$

will be computed simultaneously, and they comprise the first group of BLAS 2.5 transformations in (3.1). By simultaneous computation we mean that as soon as one component of $z_{j,k}$ is computed, $x_{j,k}^{(1)}$ is updated with this new data by the BLAS 1 `saxpy()` operation. The components of $z_{j,k}$ can be partitioned in blocks of dimension $c$, so that BLAS 2 `segmv()` is used in the simultaneous computation instead of BLAS 1 operations. This improves the cache memory usage even more.

In the $k$th step of the inner loop for the last $(g+1)$th block update with $\mathbf{V}_{g+1,k-1}$, computation of $z_{g+1,k}$ and $x_{g+1,k}^{(1)}$ will be done simultaneously. Again let $\ell = gb+k$. First, we have

$$\begin{aligned} A_{g+1,k-1}(:\,,\ell+1\colon n) &= A_{g+1,k-2}(:\,,\ell+1\colon n) \\ &\quad -A_{g+1,k-2}(:\,,\ell\colon n)v_{g+1,k-1}v_{g+1,k-1}(2\colon n-\ell+1)^T \\ &= A_{g+1,k-2}(:\,,\ell+1\colon n) - x_{g+1,k-1}^{(3)}v_{g+1,k-1}(2\colon n-\ell+1)^T, \end{aligned}$$

where $x_{g+1,k}^{(3)} = A_{g+1,k-1}(:\,,\ell+1\colon n)v_{g+1,k}$, and from (3.4) it follows that

$$\begin{aligned} &A_{g+1,k-1}(:\,,\ell+1\colon n)v_{g+1,k} \\ &\quad = \frac{\sqrt{2}[A_{g+1,k-1}(:\,,\ell+1\colon n)z_{g+1,k}-\phi_{\ell+1}A_{g+1,k-1}(:\,,\ell+1)]}{\|z_{g+1,k}-\phi_{\ell+1}e_1\|_2}. \end{aligned} \tag{3.7}$$

Again, if we define

$$x_{g+1,k}^{(1)} = A_{g+1,k-1}(:\,,\ell+1\colon n)z_{g+1,k}$$

as the first phase in the computation of the vector $x_{g+1,k}^{(3)}$, then

$$\boxed{\begin{array}{rcl} A_{g+1,k-1}(:\,,\ell+1\colon n) & = & A_{g+1,k-2}(:\,,\ell+1\colon n) \\ & & -x_{g+1,k-1}^{(3)}v_{g+1,k-1}(2\colon n-\ell+1)^T \\ z_{g+1,k} & = & A_{g+1,k-1}(:\,,\ell+1\colon n)^T u_\ell \\ x_{g+1,k}^{(1)} & = & A_{g+1,k-1}(:\,,\ell+1\colon n)z_{g+1,k} \end{array}}$$

comprises the second group of BLAS 2.5 transformations in (3.1).

The reason why these operations are performed simultaneously is that the same parts of the matrix $A$ are involved, as well as the same parts of the vector $z_{j,k}$. Thus, when a particular block of the matrix and the vector is stored in the fast cache memory, all of the operations can be done without transferring blocks from slower memory to the cache, saving some of the time spent on memory by Algorithm 2.1.

Before stating a detailed algorithm, we have to introduce one more definition. The update of the $\ell$th column of the matrix $A_{j,k}$, where $\ell = (j-1)b+k$, is done by the following relations:

$$\begin{aligned} A_{j,k}(:,\ell) &= [A_{j,0}\mathbf{V}_{j,1}\dots\mathbf{V}_{j,k-1}](:,\ell) \\ &= A_{j,0}(:,\ell) - A_{j,0}Y_j(:,1\colon k-1)W_j(\ell,1\colon k-1)^T. \end{aligned}$$

The term $A_{j,0}Y_j(:,1\colon k-1)$ also occurs in relations (3.5) and (3.6); hence, we define $X_j = A_{j,0}Y_j$. From the definition of the matrices $Y_j$ and $W_j$ in [4], $W_j$ and $X_j$ satisfy the following recurrences:

$$\begin{aligned} W_j(:,1) &= \mathbf{v}_{j,1}, \\ W_j(:,1\colon k) &= [W_j(:,1\colon k-1),\ \mathbf{v}_{j,k}], \\ X_j(:,1) &= A_{j,0}Y_j(:,1) = A_{j,0}\mathbf{v}_{j,1}, \\ X_j(:,1\colon k) &= A_{j,0}Y_j(:,1\colon k) \\ &= [X_j(:,1\colon k-1),\ A_{j,0}\mathbf{v}_{j,k} \\ &\qquad -X_j(:,1\colon k-1)W_j(:,1\colon k-1)^T\mathbf{v}_{j,k}]. \end{aligned} \tag{3.8}$$

Now we can state the complete algorithm.

ALGORITHM 3.1. [illegible] $A \in \mathbb{R}^{m\times n}$ rank$(A) = n > 2$ [illegible] $U$ [illegible] $B$ [illegible] $V$ [illegible] $A = UBV^T$

```
Initialize:
  the block dimension for aggregated Householder transformations $b$;
  the block dimension for BLAS 2.5 transformations $c$;
$A_{1,0} = A$.
$s_1 = A_{1,0}(:,1)$.
$g = \lfloor (n-2)/b \rfloor$.
for $j = 1\colon g$
  { [illegible] $j$ [illegible] $A$ [illegible] 2.5 [illegible] (3.1) }
  $X_j = 0_{m\times b}$. $W_j = 0_{n\times b}$.
  for $k = 1\colon b$
    $\ell = (j-1)b+k$.
    $A_{j,k}(:,1\colon \ell-1) = A_{j,k-1}(:,1\colon \ell-1)$.
    if $k > 1$
      $A_{j,k}(:,\ell) = A_{j,0}(:,\ell) - X_j(:,1\colon k-1)W_j(\ell,1\colon k-1)^T$.
      $s_\ell = A_{j,k}(:,\ell) - \phi_\ell u_{\ell-1}$.
    else
      $A_{j,k}(:,\ell) = A_{j,k-1}(:,\ell)$.
    end.
    $\psi_\ell = \|s_\ell\|_2$.
    $u_\ell = s_\ell/\psi_\ell$.
    if $k > 1$
      $z^{(1)}_{j,k} = -W_j(\ell+1\colon n,1\colon k-1)X_j(:,1\colon k-1)^T u_\ell$.
    else
      $z^{(1)}_{j,k} = 0_{(n-\ell)\times 1}$.
    end.
    $x^{(1)}_{j,k} = 0_{m\times 1}$.
    for $i = \ell+1 \colon c \colon n$
      $d = \min(c, n-i+1)$.
      $z_{j,k}(i-\ell \colon i-\ell+d-1) = z^{(1)}_{j,k}(i-\ell \colon i-\ell+d-1) + A_{j,0}(\colon, i \colon i+d-1)^T u_\ell$.
      $x^{(1)}_{j,k} = x^{(1)}_{j,k} + A_{j,0}(\colon, i \colon i+d-1) z_{j,k}(i-\ell \colon i-\ell+d-1)$.
    end.
    $[\phi_{\ell+1}, v_{j,k}, x^{(3)}_{j,k}] =$ householder $2(z_{j,k}, x^{(1)}_{j,k}, A_{j,0}(\colon, \ell+1))$.
    $W_j(\ell+1 \colon n, k) = v_{j,k}$.
    $x^{(4)}_{j,k} = x^{(3)}_{j,k} - X_j(\colon, 1 \colon k-1) W_j(\ell+1 \colon n, 1 \colon k-1)^T v_{j,k}$.
    $X_j(\colon, k) = x^{(4)}_{j,k}$.
  end.
  { [illegible] $A$ [illegible]
  [illegible] $j$ [illegible] }
  $A_{j+1,0}(\colon, 1 \colon jb) = A_{j,b}(\colon, 1 \colon jb)$.
  $A_{j+1,0}(\colon, jb+1 \colon n) = A_{j,b}(\colon, jb+1 \colon n) - X_j W_j(jb+1 \colon n, \colon)^T$.
  $s_{jb+1} = A_{j+1,0}(\colon, jb+1) - \phi_{jb+1} u_{jb}$.
end.
$r = n - gb$.
{ [illegible] $A$ [illegible] 2.5
[illegible] (3.1) }
for $k = 1 \colon r-1$
  $\ell = gb + k$.
  if $k > 1$
    $A_{g+1,k}(\colon, 1 \colon \ell-1) = A_{g+1,k-1}(\colon, 1 \colon \ell-1)$.
    $A_{g+1,k}(\colon, \ell) = A_{g+1,k-1}(\colon, \ell) - v_{g+1,k-1}(1) x^{(3)}_{g+1,k-1}$.
    $s_\ell = A_{g+1,k}(\colon, \ell) - \phi_\ell u_{\ell-1}$.
  else
    $A_{g+1,k}(\colon, 1 \colon \ell) = A_{g+1,k-1}(\colon, 1 \colon \ell)$.
  end.
  $\psi_\ell = \|s_\ell\|_2$.
  $u_\ell = s_\ell / \psi_\ell$.
  $x^{(1)}_{g+1,k} = 0_{m\times 1}$.
  for $i = \ell+1 \colon n$
    if $k > 1$
      $A_{g+1,k-1}(\colon, i) = A_{g+1,k-2}(\colon, i) - v_{g+1,k-1}(i-\ell+1) x^{(3)}_{g+1,k-1}$.
    end.
    if $\ell < n-1$
      $z_{g+1,k}(i-\ell) = A_{g+1,k-1}(\colon, i)^T u_\ell$.
      $x^{(1)}_{g+1,k} = x^{(1)}_{g+1,k} + z_{g+1,k}(i-\ell) A_{g+1,k-1}(\colon, i)$.
    end.
    end.
    if $\ell < n-1$
        $[\phi_{\ell+1}, v_{g+1,k}, x^{(3)}_{g+1,k}]$ = householder 2$(z_{g+1,k}, x^{(1)}_{g+1,k}, A_{g+1,k-1}(:\,,\ell+1))$.
    end .
end.
$\phi_n = u^T_{n-1} A_{g+1,r-1}(:\,,n)$.
$s_n = A_{g+1,r-1}(:\,,n) - \phi_n u_{n-1}$.
$\psi_n = \|s_n\|_2$.
$u_n = s_n/\psi_n$.
```

---

```
$V^T$ = householder_product$(v_{1,1}, \ldots, v_{g+1,r-2})$.
end.
```

The auxiliary function **householder** 2() is defined as follows.

```
function $[\phi, v, y]$=householder 2$(z, x, b)$
{ [illegible] householder 2() [illegible] $\phi$ $v$ [illegible] $y$ [illegible] $\phi$ [illegible] $v$ [illegible]
[illegible] householder() [illegible] 2.1 [illegible]
[illegible] $V = I - vv^T$ [illegible] $Vz = \phi e_1$ [illegible] $y = Bv$
[illegible] $x = Bz$ [illegible] $b = Be_1$ }
$[\phi, v]$=householder$(z)$;
if $\phi > 0$
    $w = x - \phi b$;
    $y = \sqrt{2}w/\|z - \phi e_1\|_2$;
else
    $y = 0$;
end
```

[illegible] 3.1. The choice of block dimensions $b$ and $c$ depends upon which computer executes Algorithm 3.1. Their sizes are chosen to obtain optimal efficiency. In LAPACK routines, the function *ilaenv()* is used to determine the optimal block size for block algorithms. Section 5.2 in [1] explains how *ilaenv()* works: "The version of *ilaenv()* supplied with the package contains default values that led to good behavior over a reasonable number of the test machines, but to achieve optimal performance, it may be beneficial to tune *ilaenv()* for the particular machine environment." Our optimal block dimensions were obtained through tests.

Algorithm 2.1 is numerically backward stable for computing $B$ and $V$, but what about Algorithm 3.1? The answer to this question is given by Theorem 3.4. Before stating a proof of Theorem 3.4, we will need the results of three technical lemmas. The lemmas are based on the numerical analysis of basic numerical algorithms given by Higham [25] and the analysis of the modified Gram–Schmidt algorithm given by Björck and Paige [5]. The proofs of the lemmas can be found in [7]. In our numerical analysis we will use the following notation: Tildes (˜) will mark computed quantities, and hats (ˆ, ˘ ) will denote vectors and matrices that correspond to certain exact relations and exist only as theoretical entities, not actually computed.

LEMMA 3.1. [illegible] 3.1 [illegible] $\varepsilon$ [illegible]

$$\tilde{v}_{j,k} = \hat{v}_{j,k} + \delta\hat{v}_{j,k}, \qquad \|\delta\hat{v}_{j,k}\|_2 \le O(n-l)\varepsilon,$$

$$\tilde{W}_j(:\,,1\colon k) = \hat{W}_j(:\,,1\colon k) + \delta\hat{W}_j(:\,,1\colon k), \qquad \|\delta\hat{W}_j(:\,,1\colon k)\|_F \le O(\sqrt{k}n)\varepsilon,$$

$$\tilde{X}_j(:\,,k) = \tilde{A}_{j,0}\hat{Y}_j(:\,,k) + \delta\hat{X}_j(:\,,k), \qquad \|\delta\hat{X}_j(:\,,k)\|_2 \le O(kn)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

[illegible] $\hat{v}_{j,k}$ [illegible] $\hat{V}_{j,k}$ [illegible] $\hat{W}_j$ $\hat{Y}_j$ [illegible] $\hat{X}_j = \tilde{A}_{j,0}\hat{Y}_j$ [illegible] $\hat{v}_{j,k}$ [illegible] (3.8) [illegible]

$$\|\hat{W}_j(:,1\colon k)\|_F \le \sqrt{2}\sqrt{k},$$
$$\|\hat{X}_j(:,1\colon k)\|_F = \|\tilde{A}_{j,0}\hat{Y}_j(:,1\colon k)\|_F \le 2\sqrt{2}\sqrt{k}\|\tilde{A}_{j,0}\|_F.$$

LEMMA 3.2. [illegible] $\tilde{B}$ [illegible] 3.1 [illegible]

$$\begin{bmatrix} \tilde{\phi}_{k+1}e_k + \tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} = \hat{P}_{k+1}\hat{P}_k \left( \begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right),$$
$$\left\| \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix} \right\|_2 \le O(bm)\varepsilon\|F\|_F,$$

[illegible] $\hat{P}_k$ $k = 1,\dots,n$ [illegible] $(m+n)\times(m+n)$ [illegible] [5]

LEMMA 3.3. [illegible] $\tilde{u}_\ell$ [illegible] $\tilde{A}_{g+1,r-1}(:,\ell+2\colon n)$ [illegible] 3.1 [illegible]

$$\|\tilde{u}_\ell^T\tilde{A}_{g+1,r-1}(:,\ell+2\colon n)\|_2 \le O(b^2n + bm + bn^2)\varepsilon\|A\|_F.$$

THEOREM 3.4. [illegible] $\tilde{B}$ [illegible] 3.1 [illegible] $(m+n)\times(m+n)$ [illegible] $\hat{\mathcal{P}}$ [illegible] $n\times n$ [illegible] $\hat{V}$ [illegible] $\Delta A$ $\delta A$ [illegible]

$$\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \hat{\mathcal{P}}^T \begin{bmatrix} \Delta A \\ A+\delta A \end{bmatrix} \hat{V}, \quad \left\| \begin{bmatrix} \Delta A \\ \delta A \end{bmatrix} \right\|_F \le \xi\|A\|_F, \tag{3.9}$$

[illegible] $0 \le \xi \le O(b(mn+n^3))\varepsilon$ [illegible] $\tilde{V}$ [illegible] $\hat{V}$ [illegible] $\|\tilde{V} - \hat{V}\|_F \le O(n^2)\varepsilon$ [illegible] $\breve{U}$ [illegible] $\delta\breve{A}$ [illegible]

$$A + \delta\breve{A} = \breve{U}\tilde{B}\hat{V}^T, \quad \|\delta\breve{A}\|_F \le \sqrt{2}\xi\|A\|_F. \tag{3.10}$$

[illegible]. The proof is rather technical and we will divide it into three steps.

[illegible] 1. We will set $F = \mathrm{fl}(AV) = \tilde{A}_{g+1,r-1}$, $\ell = (j-1)b + k$ and $r = n - gb$, where $\tilde{A}_{g+1,r-1}$ is the result of Algorithm 3.1 performed in finite precision arithmetic. Thus, in floating point computation we can use $f_\ell = F(:,\ell)$ instead of

$$f_\ell = \begin{cases} \tilde{A}_{j,k}(:,\ell) & \text{for } j = 1,\dots g,\ k = 1,\dots,b,\ \ell = 1,\dots,gb, \\ \tilde{A}_{g+1,k}(:,\ell) & \text{for } k = 1,\dots,r-1,\ \ell = gb+1,\dots,n-1, \\ \tilde{A}_{g+1,r-1}(:,n) & \text{for } \ell = n, \end{cases}$$

because the denoted column will not be modified in successive steps of the algorithm (see Figure 3.1).

In this step of the proof we will analyze the application of Householder reflectors to the matrix $A$, in floating point arithmetic. This application is divided into $g$ steps, where $b$ columns of $F$ are computed in each step, and $r$ remaining steps, where only one column of $F$ is computed per step. First, we are investigating the computations performed in one block $j \in \{1,2,\dots,g\}$.

Lemma 3.1 gives the error estimate

$$\tilde{X}_j(:,1\colon k) = \tilde{A}_{j,0}\hat{Y}_j(:,1\colon k) + \delta\hat{X}_j(:,1\colon k),$$

where

$$\|\delta\hat{X}_j(:,1\colon k)\|_F = \sqrt{\sum_{i=1}^{k} \|\delta\hat{X}_j(:,i)\|_2^2} \le O(k^{\frac{3}{2}}n)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

and

$$\tilde{W}_j(:,1\colon k) = \hat{W}_j(:,1\colon k) + \delta\hat{W}_j(:,1\colon k),$$

with

$$\|\delta\hat{W}_j(:,1\colon k)\|_F \le O(\sqrt{k}n)\varepsilon.$$

The only thing that remains is to find a bound on the error introduced by the application of Householder reflectors to the matrix $A$. First, for $\ell = (j-1)b + k$, $k = 1, \ldots, b$, we define the matrices $\hat{V}_{j,k}, \tilde{V}_{j,k} \in \mathbb{R}^{(n-l)\times(n-l)}$ as (see [7] for the proof of Lemma 3.1 and [1] for the documentation of the LAPACK routine `slarfg()`)

$$\hat{V}_{j,k} = I - \hat{\tau}_{j,k}\hat{v}_{j,k}\hat{v}_{j,k}^T, \quad \tilde{V}_{j,k} = I - \tilde{\tau}_{j,k}\tilde{v}_{j,k}\tilde{v}_{j,k}^T,$$

and $\hat{Q}_j, \tilde{Q}_j \in \mathbb{R}^{n\times n}$ as

$$\hat{Q}_j = \begin{bmatrix} I_{jb} & 0 \\ 0 & \hat{V}_{j,b} \end{bmatrix} \cdots \begin{bmatrix} I_{(j-1)b+2} & 0 \\ 0 & \hat{V}_{j,2} \end{bmatrix} \begin{bmatrix} I_{(j-1)b+1} & 0 \\ 0 & \hat{V}_{j,1} \end{bmatrix},$$
$$\tilde{Q}_j = \begin{bmatrix} I_{jb} & 0 \\ 0 & \tilde{V}_{j,b} \end{bmatrix} \cdots \begin{bmatrix} I_{(j-1)b+2} & 0 \\ 0 & \tilde{V}_{j,2} \end{bmatrix} \begin{bmatrix} I_{(j-1)b+1} & 0 \\ 0 & \tilde{V}_{j,1} \end{bmatrix},$$

where $\hat{Q}_j = I - \hat{W}_j\hat{Y}_j^T$ and $\tilde{Q}_j = I - \tilde{W}_j\tilde{Y}_j^T$.

Then, for $\hat{X}_j, \tilde{X}_j \in \mathbb{R}^{m\times b}$ and $\hat{W}_j, \tilde{W}_j \in \mathbb{R}^{n\times b}$, from Lemma 3.1 it follows that

$$\begin{aligned} \tilde{A}_{j+1,0} &= \mathrm{fl}(\tilde{A}_{j,0}\tilde{Q}_j^T) = \mathrm{fl}(\tilde{A}_{j,0} - \mathrm{fl}(\tilde{X}_j\tilde{W}_j^T)) \\ &= \tilde{A}_{j,0} - [(\hat{X}_j + \delta\hat{X}_j)(\hat{W}_j^T + \delta\hat{W}_j^T) + \delta_1 A_{j+1,0}] + \delta_2 A_{j+1,0} \\ &= \tilde{A}_{j,0} - \hat{X}_j\hat{W}_j^T + \delta A_{j+1,0} = \tilde{A}_{j,0} - \tilde{A}_{j,0}\hat{Y}_j\hat{W}_j^T + \delta A_{j+1,0} \\ &= \tilde{A}_{j,0}\hat{Q}_j^T + \delta A_{j+1,0}, \end{aligned}$$

where

$$\|\delta_1 A_{j+1,0}\|_F \le O(b^2)\varepsilon\|\tilde{A}_{j,0}\|_F, \qquad \|\delta_2 A_{j+1,0}\|_F \le O(b)\varepsilon\|\tilde{A}_{j,0}\|_F,$$

which implies

$$\|\delta A_{j+1,0}\|_F \le O(b^2 n)\varepsilon\|\tilde{A}_{j,0}\|_F.$$

Finally, we obtain the result for $F = \tilde{A}_{g+1,r-1}$, where the first $g$ updates are performed as shown above, and the last $r - 1 = n - gb - 1$ updates can be considered in the same framework but with $b = 1$. Let us denote $g_t = g + r - 1$ as the total number of update steps. We can note that

$$\begin{aligned}\|F\|_F &\leq \|\tilde{A}_{g+1,r-2}\|_F + O(\varepsilon) \leq \cdots \leq \|\tilde{A}_{g+1,1}\|_F + O(\varepsilon) \leq \|\tilde{A}_{g+1,0}\|_F + O(\varepsilon)\\ &\leq \|\tilde{A}_{g,0}\|_F + O(\varepsilon) \leq \cdots \leq \|\tilde{A}_{2,0}\|_F + O(\varepsilon) \leq \|A\|_F + O(\varepsilon).\end{aligned}$$

Then by induction we have

$$\begin{aligned}F &= ((\ldots((A\hat{Q}_1^T + \delta A_{2,0})\hat{Q}_2^T + \delta A_{3,0})\ldots)\hat{Q}_{g_t-1}^T + \delta A_{g+1,r-2})\hat{Q}_{g_t}^T + \delta A_{g+1,r-1}\\ &= A\hat{Q}_1^T\hat{Q}_2^T\cdots\hat{Q}_{g_t}^T + \sum_{j=1}^{g}\delta A_{j+1,0}\hat{Q}_{j+1}^T\ldots\hat{Q}_{g_t}^T + \sum_{k=1}^{r-1}\delta A_{g+1,k}\hat{Q}_{g+k+1}^T\ldots\hat{Q}_{g_t}^T\\ &= A\hat{V} + \delta_1 F,\end{aligned}$$

where

$$\|\delta_1 F\|_F \leq [O(gb^2n) + O((n-gb)n)]\varepsilon\|A\|_F \leq O(bn^2)\varepsilon\|A\|_F.$$

At the end of this step of the proof, for $\hat{V} = \hat{Q}_1^T\hat{Q}_2^T\ldots\hat{Q}_{g_t}^T$ we can state that

$$F = (A + \delta_1 A)\hat{V}, \qquad \|\delta_1 A\|_F \leq \eta_F\|A\|_F, \quad \eta_F \leq O(bn^2)\varepsilon,$$

where $\delta_1 A = \delta_1 F \cdot \hat{V}^T$.

2. Since the computation of $\tilde{B}$ from $F = [f_1, \ldots, f_n]$ corresponds to the modified Gram–Schmidt algorithm, we can use the results from [5] and represent the computation in an equivalent form, as the Householder QR factorization of the augmented matrix

$$\begin{bmatrix} 0 \\ F \end{bmatrix} = \begin{bmatrix} 0 \\ A + \delta_1 A \end{bmatrix}\hat{V}.$$

By Lemma 3.2, the following relations hold:

$$\begin{bmatrix} \tilde{\phi}_{k+1}e_k + \tilde{\psi}_{k+1}e_{k+1} \\ 0 \end{bmatrix} = \hat{P}_{k+1}\hat{P}_k\left\{\begin{bmatrix} 0 \\ f_{k+1} \end{bmatrix} + \begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix}\right\},$$

$$\left\|\begin{bmatrix} \Delta f_{k+1} \\ \delta f_{k+1} \end{bmatrix}\right\|_2 \leq O(bm)\varepsilon\|F\|_F,$$

where

$$\hat{P}_k = I_{m+n} - \begin{bmatrix} -e_k \\ \hat{u}_k \end{bmatrix}\begin{bmatrix} -e_k^T & \hat{u}_k^T \end{bmatrix},$$

and $\hat{u}_k = \tilde{s}_k/\|\tilde{s}_k\|_2$ is the exact vector with $\|\hat{u}_k\|_2 = 1$. Putting all columns of $\tilde{B}$ together, we get

$$\begin{aligned}\begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \left[\begin{bmatrix} \tilde{\psi}_1 e_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \tilde{\phi}_2 e_1 + \tilde{\psi}_2 e_2 \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} \tilde{\phi}_n e_{n-1} + \tilde{\psi}_n e_n \\ 0 \end{bmatrix}\right]\\ &= \left[\hat{P}_1\begin{bmatrix} \Delta f_1 \\ f_1 + \delta f_1 \end{bmatrix}, \hat{P}_2\hat{P}_1\begin{bmatrix} \Delta f_2 \\ f_2 + \delta f_2 \end{bmatrix}, \ldots, \hat{P}_n\hat{P}_{n-1}\begin{bmatrix} \Delta f_n \\ f_n + \delta f_n \end{bmatrix}\right],\end{aligned}$$

and using the fact that

$$\hat{P}_i \begin{bmatrix} \tilde{B}(:,j) \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{\phi}_j e_{j-1} + \tilde{\psi}_j e_j \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{B}(:,j) \\ 0 \end{bmatrix} \quad \text{for all } i \neq j, j-1,$$

we obtain

$$\begin{aligned} \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = \Bigg[ & \hat{P}_n \hat{P}_{n-1} \dots \hat{P}_2 \hat{P}_1 \begin{bmatrix} \Delta f_1 \\ f_1 + \delta f_1 \end{bmatrix}, \hat{P}_n \hat{P}_{n-1} \dots \hat{P}_2 \hat{P}_1 \begin{bmatrix} \Delta f_2 \\ f_2 + \delta f_2 \end{bmatrix}, \\ & \hat{P}_n \hat{P}_{n-1} \dots \hat{P}_3 \hat{P}_2 \begin{bmatrix} \Delta f_3 \\ f_3 + \delta f_3 \end{bmatrix}, \hat{P}_n \hat{P}_{n-1} \dots \hat{P}_4 \hat{P}_3 \begin{bmatrix} \Delta f_4 \\ f_4 + \delta f_4 \end{bmatrix}, \\ & \dots, \hat{P}_n \hat{P}_{n-1} \hat{P}_{n-2} \begin{bmatrix} \Delta f_{n-1} \\ f_{n-1} + \delta f_{n-1} \end{bmatrix}, \hat{P}_n \hat{P}_{n-1} \begin{bmatrix} \Delta f_n \\ f_n + \delta f_n \end{bmatrix} \Bigg]. \end{aligned}$$

The $k$th column of the computed bidiagonal matrix is of the form

$$\hat{P}_n \hat{P}_{n-1} \dots \hat{P}_k \hat{P}_{k-1} \begin{bmatrix} \Delta f_k \\ f_k + \delta f_k \end{bmatrix},$$

and the desired form is

$$\hat{P}_n \hat{P}_{n-1} \dots \hat{P}_2 \hat{P}_1 \begin{bmatrix} \hat{\Delta} f_k \\ f_k + \hat{\delta} f_k \end{bmatrix} = \hat{\mathcal{P}}^T \begin{bmatrix} \hat{\Delta} f_k \\ f_k + \hat{\delta} f_k \end{bmatrix}, \quad \mathcal{P} = \hat{P}_1 \hat{P}_2 \dots \hat{P}_{n-1} \hat{P}_n.$$

The first two columns ($k = 1, 2$) are already in the desired form and $\hat{\Delta} f_k = \Delta f_k$, $\hat{\delta} f_k = \delta f_k$. For $k \geq 3$ we write

$$\begin{bmatrix} \tilde{B}(:,k) \\ 0 \end{bmatrix} = (\hat{P}_n \hat{P}_{n-1} \dots \hat{P}_k \hat{P}_{k-1} \overbrace{\hat{P}_{k-2} \dots \hat{P}_2 \hat{P}_1)(\hat{P}_1 \hat{P}_2 \dots \hat{P}_{k-2})}^{I} \begin{bmatrix} \Delta f_k \\ f_k + \delta f_k \end{bmatrix},$$

and then

$$\begin{aligned} \hat{P}_1 \hat{P}_2 \dots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ f_k + \delta f_k \end{bmatrix} &= \begin{bmatrix} 0 \\ f_k \end{bmatrix} + \begin{bmatrix} \Delta_1 f_k \\ \delta_1 f_k \end{bmatrix} + \hat{P}_1 \begin{bmatrix} \Delta_2 f_k \\ \delta_2 f_k \end{bmatrix} + \hat{P}_1 \hat{P}_2 \begin{bmatrix} \Delta_3 f_k \\ \delta_3 f_k \end{bmatrix} \\ &\quad + \dots + \hat{P}_1 \dots \hat{P}_{k-3} \begin{bmatrix} \Delta_{k-2} f_k \\ \delta_{k-2} f_k \end{bmatrix} + \hat{P}_1 \dots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ f_k \end{bmatrix} + \begin{bmatrix} \hat{\Delta} f_k \\ \hat{\delta} f_k \end{bmatrix}, \end{aligned}$$

where

$$\begin{bmatrix} \Delta_j f_k \\ \delta_j f_k \end{bmatrix} = \begin{bmatrix} e_j \\ -\hat{u}_j \end{bmatrix} (\hat{u}_j^T f_k), \quad j = 1, \dots, k-2,$$

with

$$\begin{bmatrix} \hat{\Delta} f_k \\ \hat{\delta} f_k \end{bmatrix} = \hat{P}_1 \cdots \hat{P}_{k-2} \begin{bmatrix} \Delta f_k \\ \delta f_k \end{bmatrix} + \begin{bmatrix} \Delta_1 f_k \\ \delta_1 f_k \end{bmatrix} + \sum_{j=2}^{k-2} \hat{P}_1 \dots \hat{P}_{j-1} \begin{bmatrix} \Delta_j f_k \\ \delta_j f_k \end{bmatrix}.$$

Hence,

$$\begin{aligned} \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} &= \hat{\mathcal{P}}^T \left[ \begin{bmatrix} \hat{\Delta} f_1 \\ f_1 + \hat{\delta} f_1 \end{bmatrix}, \dots, \begin{bmatrix} \hat{\Delta} f_k \\ f_k + \hat{\delta} f_k \end{bmatrix}, \dots, \begin{bmatrix} \hat{\Delta} f_n \\ f_n + \hat{\delta} f_n \end{bmatrix} \right] \\ &= \hat{\mathcal{P}}^T \left\{ \begin{bmatrix} 0 \\ F \end{bmatrix} + \begin{bmatrix} \Delta F \\ \delta F \end{bmatrix} \right\}, \end{aligned}$$

where, after suitable reordering of the entries in the sums,

$$\begin{bmatrix}\Delta F\\ \delta F\end{bmatrix} = \left[\begin{bmatrix}\Delta f_1\\ \delta f_1\end{bmatrix}, \begin{bmatrix}\Delta f_2\\ \delta f_2\end{bmatrix}, \dots, \hat{P}_1\dots\hat{P}_{k-2}\begin{bmatrix}\Delta f_k\\ \delta f_k\end{bmatrix}, \dots, \hat{P}_1 \dots \hat{P}_{n-2}\begin{bmatrix}\Delta f_n\\ \delta f_n\end{bmatrix}\right]$$
$$+ \sum_{j=1}^{n-2}\hat{P}_1\dots\hat{P}_{j-1}\left[\underbrace{0,\ \dots\ ,0,}_{j+1}\ \begin{bmatrix}\Delta_j f_{j+2}\\ \delta_j f_{j+2}\end{bmatrix},\dots,\begin{bmatrix}\Delta_j f_n\\ \delta_j f_n\end{bmatrix}\right].$$

Taking norms, we obtain

$$\begin{aligned}\left\|\begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\right\|_F &\leq O(bm\sqrt{n})\varepsilon\|F\|_F + \sqrt{2}\sum_{j=1}^{n-2}\|\hat{u}_j^T\begin{bmatrix} f_{j+2} & f_{j+3} & \dots & f_n\end{bmatrix}\|_2\\ &\leq O(bm\sqrt{n})\varepsilon\|F\|_F + \sqrt{2}\sum_{j=1}^{n-2}(\|\tilde{u}_j^T\tilde{A}_{g+1,r-1}(:,j+2\colon n)\|_2\\ &\quad +\|\delta\hat{u}_j\|_2\|F(:,j+2\colon n)\|_F).\end{aligned}$$

It remains to estimate the products $\tilde{u}_i^T f_\ell$ for $\ell = 3,\dots,n$ and $i = 1,\dots,\ell-2$, where $\ell = (j-1)b+k$, $k = 1,\dots,b$. For this estimate, the important role plays the choice of the vector $z_{j,k}$. From Lemma 3.3 it follows that

$$\|\tilde{u}_\ell^T\tilde{A}_{g+1,r-1}(:,\ell+2\colon n)\|_2 \leq O(b^2n + bm + bn^2)\varepsilon\|A\|_F.$$

Then

$$\left\|\begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\right\|_F \leq O(b(mn+n^3))\varepsilon\|F\|_F \leq O(b(mn+n^3))(1+\eta_F)\|A\|_F.$$

To get the relation (3.9), we collect the perturbations from both implicit tridiagonalization and the Gram-Schmidt computation

$$\begin{aligned}\begin{bmatrix}\tilde{B}\\ 0\end{bmatrix} &= \hat{\mathcal{P}}^T\left\{\begin{bmatrix}0\\ F\end{bmatrix} + \begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\right\} = \hat{\mathcal{P}}^T\left\{\begin{bmatrix}0\\ A+\delta_1 A\end{bmatrix}\hat{V} + \begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\right\}\\ &= \hat{\mathcal{P}}^T\left\{\begin{bmatrix}0\\ A+\delta_1 A\end{bmatrix} + \begin{bmatrix}\Delta F\\ \delta F\end{bmatrix}\hat{V}^T\right\}\hat{V}.\end{aligned}$$

3. Finally, using $\mathcal{P}_{11} = \mathcal{P}(1\colon n, 1\colon n)$, $\mathcal{P}_{21} = \mathcal{P}(n+1\colon n+m, 1\colon n)$, we have

$$\begin{bmatrix}\Delta A\\ A+\delta A\end{bmatrix}\hat{V} = \begin{bmatrix}\mathcal{P}_{11}\\ \mathcal{P}_{21}\end{bmatrix}\tilde{B},\ \ \mathcal{P}_{11}^T\mathcal{P}_{11} + \mathcal{P}_{21}^T\mathcal{P}_{21} = I,$$

and (3.10) follows by an application of [5, Lemma 3.1]. The proof that (3.10) holds for the nonblock version of the algorithm is given in Theorem 3.18 [3]. The same arguments can be applied to the block version.

Note that in (3.10) we can write $A + \breve{\delta A} = \breve{U}\tilde{B}\tilde{V}^T(I+\Gamma)$, $\|\Gamma\|_F \leq O(n^2)\varepsilon$. □

In our numerical experiments, the optimal choice for the block dimension $b$ was usually 16, so the accuracy predicted by the bound in Theorem 3.4 is close to the accuracy predicted by Theorem 2.1.

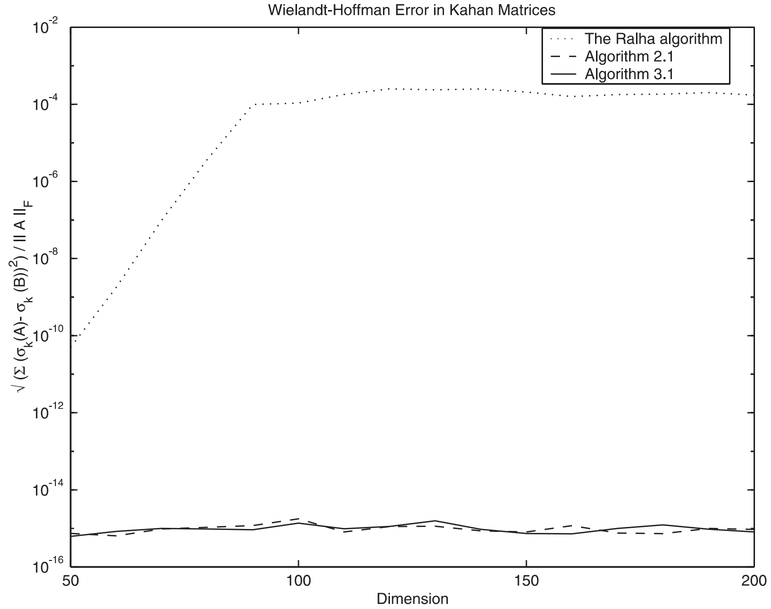


FIG. 3.2. *Error in singular values from Example* 3.1.

[illegible] 3.1. Let $A = [a_{ij}]$ be the $n \times n$ Kahan matrix as in [3], with

$$a_{ij} = \begin{cases} \alpha^{i-1} & i = j, \\ -\alpha^{j-1}\beta & i > j, \end{cases}$$

where $\alpha^2 + \beta^2 = 1$ and $\alpha, \beta > 0$. For our tests we chose $\alpha = \sin(1.2)$ and $n = 50, 60, \ldots, 200$. In this case the matrices are ill-conditioned. The first $n-1$ singular values gradually decay and are bounded away from zero, but, on the other hand, the smallest singular value decays rapidly with $n$.

We compare the accuracy of Algorithm 3.1 with Algorithm 2.1 and Ralha's one-sided bidiagonalization, by using the Wielandt–Hoffman measure

$$\frac{\sqrt{\displaystyle\sum_{k=1}^{n}(\sigma_k(A) - \sigma_k(B))^2}}{\|A\|_F}. \tag{3.11}$$

This example is performed in MATLAB and in double precision. The one-sided bidiagonalization routines are implemented in the same way as the FORTRAN routines in the next section. The singular values $\sigma_k(A)$ of the matrix $A$ are computed by the MATLAB command `svd()`. The results are shown in Figure 3.2.

We can note that Algorithm 3.1 sometimes produces the bidiagonal matrix $B$ with slightly less accurate singular values than Algorithm 2.1. Theorem 3.4 asserts that the bound on (3.11) for Algorithm 3.1 is $b$ times larger than the corresponding bound for Algorithm 2.1, where $b$ is the block dimension. In our case, we took $b = 16$. If we compare the computed errors measured by (3.11), we can see that the largest difference is obtained for $n = 180$, where the error of Algorithm 3.1 is 1.67 times larger than the error of Algorithm 2.1. In this case, the estimation of the error bounds on

(3.11) from Theorems 2.1 and 3.4 are

| Algorithm 2.1 | Algorithm 3.1 |
|---|---|
| $(n^2+n^3)\varepsilon = 6.51 \cdot 10^{-10}$ | $b(n^2+n^3)\varepsilon = 1.04 \cdot 10^{-8}$ |

Hence, our computed results are in accordance with the approximate bounds in Theorems 2.1 and 3.4 and reveal that these bounds can significantly overestimate the actual error.

Similar to the results of Björck and Paige in [5], Corollary 3.18 in [3] states that for the nonblock version of one-sided bidiagonalization algorithm we have

$$\|\tilde{U}^T\tilde{U} - I\|_F \leq p(m,n)\kappa_2(\tilde{B})\varepsilon + O(\varepsilon^2), \tag{3.12}$$

where $\tilde{U}$ is the computed matrix of left singular vectors, $p(m,n)$ is a polynomial with modest degree, and $\kappa_2(\tilde{B}) = \|\tilde{B}\|_2\|\tilde{B}^{-1}\|_2$. In fact, arguments given in the proof of this corollary hold for any bidiagonalization algorithm for which Theorem 3.4 can be proved. That means that if $B$ is ill-conditioned, $\tilde{U}$ might be far from left orthogonal matrix, but sometimes the bound in (3.12) is too pessimistic.

On the other hand, suppose that a numerically backward stable SVD is performed on the matrix $\tilde{B}$ from Theorem 3.4, obtaining matrices $\tilde{Y}$, $\tilde{\Sigma}$, and $\tilde{W}$. From [3, Theorem 3.7] it follows that there exists a perturbation $\hat{\delta A}$ such that $A + \hat{\delta A} = \tilde{P}\tilde{\Sigma}\tilde{Q}^T$, where $\tilde{P} = \tilde{U}\tilde{Y}$, $\tilde{Q} = \tilde{V}\tilde{W}$, and $\|\hat{\delta A}\|_F \leq O(\varepsilon)\|A\|_F$. $\tilde{P}$ can be considered as a matrix of computed left singular vectors of $A$, $\tilde{Q}$ as a matrix of computed right singular vectors of $A$, and $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n)$, where $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$, is a matrix of computed singular values. By the same argument as in Corollary 3.20 in [3], we can state that

$$\|\tilde{P}_1^T\tilde{P}_1 - I\|_F \leq q(m,n)\frac{\tilde{\sigma}_1}{\tilde{\sigma}_k}\varepsilon + O(\varepsilon^2),$$

where $\tilde{P}_1$ is a matrix which contains $k$ computed left singular vectors that correspond to the $k$ largest singular values, and $q(m,n)$ is a polynomial with modest degree. This shows that the basis for the leading principal subspace of left singular vectors can be computed with near orthogonality.

## 4. Efficiency of the stable block one-sided bidiagonalization.

For the block version of the new stable one-sided bidiagonalization, extensive testing was carried out. Computations were performed in the Advanced Computing Laboratory of the Department of Mathematics, University of Zagreb. The laboratory consists of 20 computers, connected in a local 1Gb network. The specifications of the computers are shown in Table 4.1.

The computers are working under a Debian GNU/Linux operating system. The tests were written in FORTRAN 77 programming language, and a GNU (v0.5.24)

Table 4.1
*The specifications of the computers in Advanced Computing Laboratory.*

| 2 processors | Athlon Mp 1800+ |
|---|---|
| Frequency | 1533 MHz |
| L1 Cache | 64 kB |
| L2 Cache | 256 kB |
| RAM | 1 GB |

Table 4.2
*Average execution times for full SVD algorithms.*

| $m$ | $\times$ | $n$ | $t_1$ | $t_2$ | $t_L$ | $p_{2,1}$ | $p_{2,L}$ |
|---|---|---|---|---|---|---|---|
| 100 | × | 100* | 0.008 | 0.008 | 0.011 | 0.00% | 27.27% |
| 200 | × | 200* | 0.058 | **0.055** | 0.065 | 5.17% | 15.38% |
| 500 | × | 50* | 0.005 | 0.005 | 0.008 | 0.00% | 37.50% |
| 500 | × | 100* | 0.022 | **0.021** | 0.035 | 4.55% | 40.00% |
| 500 | × | 500 | 4.910 | **4.360** | 4.590 | 11.20% | 5.01% |
| 1000 | × | 100* | 0.072 | **0.054** | 0.096 | 25.00% | 43.75% |
| 1000 | × | 500 | 6.980 | 5.810 | **5.290** | 16.76% | -9.83% |
| 1000 | × | 500** | 5.550 | **5.030** | 5.290 | 9.37% | 4.91% |
| 1000 | × | 1000 | 39.810 | **35.200** | 36.730 | 11.58% | 4.17% |
| 2000 | × | 200 | 1.860 | 1.490 | **0.670** | 19.89% | -122.39% |
| 2000 | × | 200** | 0.600 | **0.590** | 0.670 | 1.67% | 11.94% |
| 2000 | × | 1000 | 58.280 | 49.630 | **44.480** | 14.84% | -11.58% |
| 2000 | × | 1000** | 47.590 | **43.150** | 44.480 | 9.33% | 2.99% |
| 2000 | × | 2000 | 320.640 | **285.960** | 294.380 | 10.82% | 2.86% |
| 3000 | × | 3000 | 1284.300 | **1152.940** | 1193.340 | 10.23% | 3.39% |

compiler with optimization (option "-O") was used to obtain executable files. LAPACK and BLAS routines were called in these programs, and ATLAS (Automatically Tuned Linear Algebra Software) tuned for Athlon processors was used to implement BLAS routines. Tests were done in single precision. Matrices in the tests were generated as products $A = U\Sigma V^T$, where $\Sigma$ is a diagonal matrix with fixed singular values $\{1, 2, \dots, n\}$, and $U$ and $V$ are random orthogonal matrices.

The block dimensions $b$ and $c$ in the tests were chosen to obtain the best execution time. In our case it turned out to be $b = 16$, and we took $c = 8$. Table 4.2 gives average execution times for full SVD algorithms, expressed in seconds.

The meaning of the headers in Table 4.2 are as follows:

$t_1$ — the SVD with Algorithm 2.1 for bidiagonalization. The LAPACK routine `sbdsqr()` is used for the SVD of a bidiagonal matrix, which implements the bidiagonal QR algorithm.

$t_2$ — the SVD with Algorithm 3.1 for bidiagonalization. The LAPACK routine `sbdsqr()` is used for the SVD of a bidiagonal matrix, which implements the bidiagonal QR algorithm.

$t_L$ — the LAPACK `sgesvd()` routine.

$p_{2,1} = 100(t_1 - t_2)/t_1$ — the percentage of time decrease, when the SVD with Algorithm 3.1 is compared to the SVD with Algorithm 2.1.

$p_{2,L} = 100(t_L - t_2)/t_L$ — the percentage of time decrease, when the SVD with Algorithm 3.1 is compared to the LAPACK routine.

* — the time was measured for 10 consecutive executions of the routines and then divided by 10.

** — QR factorization is performed before the SVD with Algorithms 2.1 and 3.1 for bidiagonalization.

We can conclude that the block version of the one-sided bidiagonalization algorithm did decrease the execution time of Algorithm 2.1, as expected. Compared to the SVD with Algorithm 2.1 the most significant time decrease is 25.00% for matrix dimensions 1000 × 100. The SVD routine with Algorithm 3.1 produces a code that is not slower than the LAPACK `sgesvd()` routine in most cases when all of the SVD factors are required, although this varies with the dimensions of the matrix. In many cases we observed some gains in speed. Here it should be emphasized that the QR factorization was not applied before the one-sided bidiagonalization algorithms except in cases denoted by $^{**}$, and we can note that the one-sided bidiagonalization algorithms achieved the largest speedup compared to the LAPACK routine when $m$ is much larger than $n$. This can be explained by the fact that the SVD with the one-sided bidiagonalization algorithm has a smaller operation count than the LAPACK `sgesvd()` routine (see [3, Table 1]). On the other hand, for dimensions $1000 \times 500$, $2000 \times 200$, and $2000 \times 1000$ the QR factorization was necessary to make Algorithm 3.1 faster than the `sgesvd()` routine. It seems that the memory hierarchy was utilized better that way. If the matrix $U$ is not needed, then the advantage of the one-sided bidiagonalization over the LAPACK routine might be lost. That happens because $U$ is always computed, whether it is needed or not (see [3, Table 1]). When solving the problems described in [8], our algorithm would be preferable since possible loss of orthogonality of the matrix $\tilde{U}$ is irrelevant in these cases, and the SVD with the one-sided bidiagonalization algorithm is often faster than the LAPACK routine.

**5. Parallel version of the new stable one-sided bidiagonalization.** The parallel bidiagonalization algorithm is performed on several processors simultaneously. Each matrix is distributed over the memories of all processors, and this distribution is balanced. This means that the dimensions of the submatrices assigned to each processor are almost the same. It is important to minimize communication between processors, as the time spent for communication can be expected to be a significant part of the overall execution time.

In our case we used the following setting:

- the processors were organized in linear order:

$$\boxed{\;1\;} \longleftrightarrow \boxed{\;\mathbf{2}\;} \longleftrightarrow \boxed{\;\mathbf{3}\;} \longleftrightarrow \boxed{\;\mathbf{4}\;};$$

- we used ScaLAPACK [6] for the computation;
- we used the message passing interface [22] for the interprocessor communication.

The matrix distribution over the processors is performed rowwise, because the algorithm is one-sided and column-oriented.

The most important features of the parallel version of the stable one-sided bidiagonalization algorithm are the following:

1. The matrix layout is one-dimensional block-cyclic row distribution. Each $m \times n$ matrix is divided in $m_b \times n$ blocks of contiguous rows, where $m_b$ is the block row dimension. Then the blocks are distributed across the processors in cyclic order, which guarantees good load balancing (see Figure 5.1 and [6]).
2. The algorithm is performed in the same way as Algorithm 2.1, with extra interprocessor communication. Interprocessor communication is required for:
    - computation of $z_k$ as matrix–vector multiplication,
    - broadcasting the vector $z_k$ to all processors,
    - computing scalar products.

   The rest of the computations consist of BLAS 1 operations (operations with vectors), as well as computation and application of Householder reflectors, which need no additional communication.

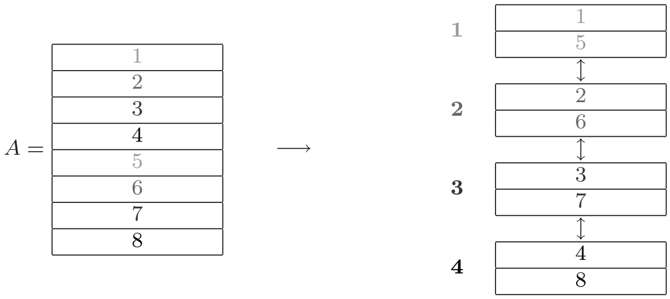


FIG. 5.1. *The block distribution of the matrix A.*

The complete parallel algorithm with explanations is listed in Algorithm 5.1.

ALGORITHM 5.1. [illegible] $A \in \mathbb{R}^{m\times n}$ rank$(A) = n > 2$ [illegible] $U = [u_1, \dots, u_n]$ [illegible] $B$ [illegible] $V$ [illegible] $A = UBV^T$

```
(1) Distribute $\Psi = [\ \psi_1 \ \dots \ \psi_n\ ]^T$ over the processors;
(2) distribute $\Phi = [\ \phi_1 \ \dots \ \phi_{n-1}\ ]^T$ over the processors;
(3) $A_0 = A$.
(4) $f_1 = A(:,1)$. $\psi_1 = \|f_1\|_2$.                         [illegible]
(5) $u_1 = f_1/\psi_1$.                                          [illegible] 1 [illegible]
                                                                 [illegible]
for $k = 1: n-2$
    (6) $z_k = A_{k-1}(:, k+1: n)^T u_k$.                        [illegible]
    (7) $[\phi_{k+1}, v_k] = $ householder$(z_k)$.                [illegible] $v_k$ [illegible] $\phi_{k+1}$.
    (8) $A_k(:, 1: k) = A_{k-1}(:, 1: k)$.
    (9) $A_k(:, k+1: n) = A_{k-1}(:, k+1: n) - A_{k-1}(:, k+1: n)v_k v_k^T$.
                                                                 [illegible]
    (10) $f_{k+1} = A_k(:, k+1)$.
    (11) $s_{k+1} = f_{k+1} - \phi_{k+1}u_k$.                    [illegible] 1 [illegible]
                                                                 [illegible]
    (12) $\psi_{k+1} = \|s_{k+1}\|_2$.                            [illegible]
    (13) $u_{k+1} = s_{k+1}/\psi_{k+1}$.                          [illegible] 1 [illegible]
                                                                 [illegible]
end.
(14) $f_n = A_{n-2}(:, n)$. $\phi_n = u_{n-1}^T f_n$.             [illegible]
(15) $s_n = f_n - \phi_n u_{n-1}$.                                [illegible] 1 [illegible]
                                                                 [illegible]
(16) $\psi_n = \|s_n\|_2$.                                        [illegible]
(17) $u_n = s_n/\psi_n$.                                          [illegible] 1 [illegible]
                                                                 [illegible]
(18) $V^T = $ householder_product$(v_1, \dots, v_{n-2})$           [illegible]
end.
```

The parallel version of the stable bidiagonalization algorithm performs the same operations as the serial nonblock version. Preliminary numerical experiments showed that a parallel block version, applying Algorithm 3.1 in parallel, has a large overhead on our computers; thus, it was almost always slower than the ScaLAPACK routine. The results of Theorem 2.1 hold for Algorithm 5.1 as well.

## 6. Efficiency of the stable parallel one-sided bidiagonalization.

**6.1. Tests performed at the Advanced Computing Laboratory.** The tests for the parallel version of the new stable bidiagonalization algorithm were done over a large variety of matrix dimensions. The computations were performed in the computational environment described in section 4 and matrices were generated in the same way. QR factorization was not performed before bidiagonalization, because Algorithm 5.1 is suitable for the parallel computing in its original form. The QR factorization would just increase the interprocessor communication. The tests confirmed this statement even for the ScaLAPACK `psgesvd()` routine, where in most cases the QR factorization introduced before bidiagonalization caused a slowdown in producing the full SVD. The block cyclic distribution was also used for testing the ScaLAPACK routine, but the linear layout of the processors may not always be optimal for this routine. So in this case we performed our test with all possible layouts for the fixed processor number, and we chose the best execution time. The parallel tests were performed over a variety of block dimensions, where block column and row dimensions were the same. The block dimensions were chosen from the set $\{8, 16, 32, 64, 128, 256\}$, and the best execution time among block dimensions was displayed in Table 6.1. For the SVD with Algorithm 5.1 the block dimension was equal to 8 in most cases, and for the ScaLAPACK SVD routine it was equal to 32 in most cases. Table 6.1 gives the average execution times expressed in seconds for full SVD algorithms when computed on $p$ processors. The meaning of the headers in Table 6.1 are as follows:

$t_3$ — the parallel SVD with Algorithm 5.1.
$p_m \times p_n$ — processor layout with the best execution time of the ScaLAPACK routine.
$t_S$ — the ScaLAPACK `psgesvd()` routine.
$p_{3,S} = 100(t_S - t_3)/t_S$ — the percentage of time decrease, when the parallel SVD with Algorithm 5.1 is compared to the ScaLAPACK routine.
$\eta_{3,p} = (t_2/t_3)/p$ — the efficiency of the parallel SVD with Algorithm 5.1 on $p$ processors.
$\eta_{S,p} = (t_L/t_S)/p$ — the efficiency of the ScaLAPACK routine on $p$ processors.

As we can see from Table 6.1, we accomplished a considerable decrease in execution time for $m \times n$ matrices when $m > n$. In that case the SVD with the described parallel version of the one-sided bidiagonalization algorithm is much faster than the ScaLAPACK routine `psgesvd()`. Compared to the ScaLAPACK routine, the most significant time decrease is 68.28% for matrix dimensions $5000 \times 100$ and for 8 processors. On the other hand, for squared matrices the ScaLAPACK routine is up to 55.08% faster, and in this case a block version of the parallel one-sided algorithm is required. Efficiency of such block algorithms is described in the next subsection.

Another important feature of parallel algorithms is the efficiency. In an ideal situation an algorithm executed on $p$ processors should be $p$ times faster than the same algorithm executed on only one processor. The efficiency measures departure from the ideal execution time. Table 6.1 shows the efficiency for both SVD algorithms applied

Table 6.1
*Average execution times for full parallel SVD algorithms.*

| $m\times n$ | $p$ | $t_3$ | $p_m \times p_n$ | $t_S$ | $p_{3,S}$ | $\eta_{3,p}$ | $\eta_{S,p}$ |
|---|---|---|---|---|---|---|---|
| 1000×100 | 4 | **0.1305** | 2×2 | 0.3004 | 56.56 | 0.1034 | 0.0799 |
| 1000×500 | 4 | **2.2733** | 2×2 | 3.1061 | 26.81 | 0.5532 | 0.4258 |
| 1000×1000 | 4 | **10.1454** | 2×2 | 12.3271 | 17.70 | 0.8674 | 0.7449 |
| 1000×1000 | 8 | **7.4465** | 8×1 | 16.0945 | 53.73 | 0.5909 | 0.2853 |
| 1000×1000 | 16 | **5.2669** | 4×4 | 8.5679 | 38.53 | 0.4177 | 0.2679 |
| 2000×200 | 4 | **0.5734** | 2×2 | 0.9863 | 41.86 | 0.2572 | 0.1698 |
| 2000×1000 | 4 | **14.1258** | 2×2 | 15.8176 | 10.70 | 0.7637 | 0.7030 |
| 2000×1000 | 8 | **9.6341** | 8×1 | 18.5550 | 48.08 | 0.5599 | 0.2996 |
| 2000×1000 | 16 | **6.6574** | 4×4 | 11.7242 | 43.22 | 0.4051 | 0.2371 |
| 2000×2000 | 4 | **92.2496** | 2×2 | 96.6973 | 4.60 | 0.7750 | 0.7611 |
| 2000×2000 | 8 | **41.3098** | 8×1 | 63.7957 | 35.25 | 0.8653 | 0.5768 |
| 2000×2000 | 16 | **35.3350** | 4×4 | 37.7961 | 6.51 | 0.5058 | 0.4868 |
| 4000×200 | 8 | **0.9272** | 8×1 | 2.4123 | 61.56 | — | — |
| 4000×1000 | 8 | **14.0244** | 8×1 | 21.7543 | 35.53 | — | — |
| 4000×1000 | 16 | **9.0173** | 16×1 | 16.2978 | 44.67 | — | — |
| 4000×4000 | 8 | **378.1784** | 8×1 | 433.7231 | 12.81 | — | — |
| 4000×4000 | 16 | 246.8128 | 2×8 | **181.0293** | -36.34 | — | — |
| 5000×100 | 8 | **0.4337** | 8×1 | 1.3674 | 68.28 | — | — |
| 5000×1000 | 16 | **10.0963** | 16×1 | 17.1380 | 41.09 | — | — |
| 5000×5000 | 16 | 538.0723 | 4×4 | **346.9633** | -55.08 | — | — |
| 8000×1000 | 16 | **13.3496** | 16×1 | 19.1207 | 30.18 | — | — |
| 8000×8000 | 16 | 2449.5129 | 4×4 | **2077.7632** | -17.89 | — | — |
| 10000×1000 | 16 | **15.4075** | 16×1 | 19.4387 | 20.74 | — | — |
| 10000×10000 | 16 | 3422.0708 | 4×4 | **2963.5295** | -15.47 | — | — |

to matrices with small dimensions. In the case of larger dimensions we were not able to run the codes on a single processor due to memory limitations, and therefore the efficiency is not computed. We can see that the parallel SVD with Algorithm 5.1 has better efficiency than the ScaLAPACK routine `psgesvd()` in all cases. The new algorithm has also better scalability than the ScaLAPACK routine when the number of processors is increased from 4 to 8 processors, which is illustrated in Figure 6.1. The $y$ axis in Figure 6.1 represents the reduction factor in execution time when the number of processors is doubled and the matrix dimensions are fixed. The labels on the $x$ axis denote matrix dimensions and ratios $p_1/p_2$, which indicate that the number of processors is increased from $p_1$ to $p_2$. Theoretically, the reduction factor in the execution time, when the number of processors is doubled, is bounded by 2. However, in two cases we obtained a reduction factor greater than 2, once for the one-sided bidiagonalization and once for the ScaLAPACK routine. This "superlinear speedup" is usually explained by the fact that, as we reduce the amount of data per processor, a larger percentage of the local data is stored in the caches, thus reducing the memory traffic overheads. So it appears that, in this case, such savings in time are more than enough to pay for the extra time required by the communication involving twice as many processors. We can conclude that, for squared matrices in case the number of processors is increased from 8 to 16 processors, the ScaLAPACK routine becomes much more efficient since it is a blocked algorithm. This illustrates the importance of the block algorithms and gives us a motivation for developing an efficient block parallel version of the one-sided bidiagonalization algorithm.

**6.2. Tests performed at the High Performance Computing Center North.** The same tests as in subsection 6.1 were also performed on the "Sarek" cluster at the High Performance Computing Center North Sweden. The cluster con-

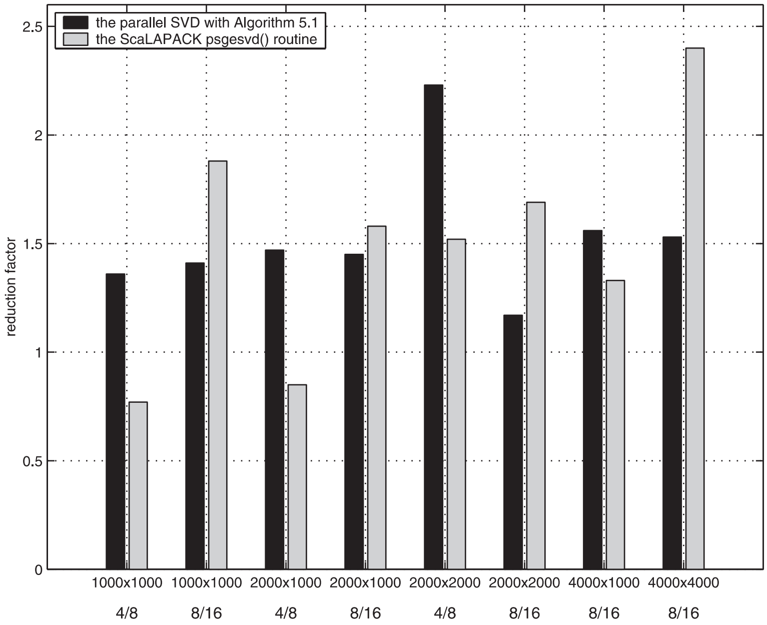


FIG. 6.1. *Reduction in execution time in the case when the number of processors is doubled.*

TABLE 6.2
*The specifications of the nodes in the "Sarek" cluster.*

| 2 processors | AMD Opteron 248 |
|---|---|
| Frequency | 2.2 GHz |
| L1 Cache | 64 kB + 64 kB |
| L2 Cache | 1024 kB |
| RAM | 8 GB |

sists of 190 nodes, connected to a Gigabit ethernet communication network and to a Myrinet high-performance network. The specifications of the nodes are shown in Table 6.2. The nodes are working under a Debian GNU/Linux operating system. The test codes were compiled with MPIF77 1.2.5.12 and PGF77 with optimization (option "-fast"). "Goto BLAS" (r0.94) was used to implement BLAS routines. QR factorization was not performed before bidiagonalization. Table 6.3 gives the execution times expressed in seconds for full SVD algorithms when computed on $p$ processors. Four rounds of tests were performed with four different parallel variants of the stable one-sided bidiagonalization algorithm. In each round the one-sided bidiagonalization routine was executed together with the ScaLAPACK `psgesvd()` routine for comparison. Due to the lack of space in Table 6.3, instead of the execution time of the ScaLAPACK routine $t_S$ we present the ratio $t_i/t_S$, where $t_i$ is the execution time of the $i$th version of the parallel bidiagonalization algorithm.

The meaning of the headers in Table 6.3 are as follows:

| | | |
|---|---|---|
| $t_S$ | — | the ScaLAPACK `psgesvd()` routine. |
| $t_3$ | — | the parallel SVD with Algorithm 5.1. |
| $t_4$ | — | the parallel SVD with a modified one-sided bidiagonalization algorithm. Campos et al. [9] have proposed a modification of the stable one-sided bidiagonalization. This version reduces communications events |

Table 6.3
*Average execution times for full parallel SVD algorithms.*

| $m{\times}n$ | $p$ | $t_3$ | $t_3/t_S$ | $t_4$ | $t_4/t_S$ | $t_5$ | $t_5/t_S$ | $t_6$ | $t_6/t_S$ |
|---|---|---|---|---|---|---|---|---|---|
| 1000×100 | 4 | **0.023** | 0.88 | **0.022** | 0.83 | **0.023** | 0.88 | **0.021** | 0.80 |
| 1000×500 | 4 | **0.530** | 0.87 | **0.524** | 0.86 | **0.495** | 0.81 | **0.503** | 0.82 |
| 1000×1000 | 4 | **3.081** | 0.84 | **3.058** | 0.83 | **2.970** | 0.81 | **2.986** | 0.81 |
| 1000×1000 | 8 | **1.732** | 0.83 | **1.697** | 0.81 | **1.669** | 0.80 | **1.641** | 0.79 |
| 1000×1000 | 16 | **1.284** | 0.92 | **1.185** | 0.85 | **1.262** | 0.91 | **1.174** | 0.85 |
| 2000×200 | 4 | **0.116** | 0.92 | **0.113** | 0.89 | **0.108** | 0.85 | **0.107** | 0.84 |
| 2000×1000 | 4 | **4.626** | 0.99 | **4.588** | 0.99 | **4.139** | 0.90 | **4.213** | 0.90 |
| 2000×1000 | 8 | **2.395** | 0.98 | **2.285** | 0.93 | **2.281** | 0.93 | **2.200** | 0.90 |
| 2000×1000 | 16 | **1.519** | 0.94 | **1.440** | 0.89 | **1.483** | 0.92 | **1.388** | 0.87 |
| 2000×2000 | 4 | **26.95** | 0.96 | **26.31** | 0.94 | **24.44** | 0.88 | **25.06** | 0.89 |
| 2000×2000 | 8 | **13.78** | 0.92 | **13.66** | 0.91 | **14.22** | 0.91 | **12.85** | 0.86 |
| 2000×2000 | 16 | **7.666** | 0.95 | **7.417** | 0.93 | **7.539** | 0.93 | **7.260** | 0.91 |
| 4000×200 | 8 | **0.131** | 0.93 | **0.126** | 0.89 | **0.123** | 0.87 | **0.116** | 0.82 |
| 4000×1000 | 8 | 4.016 | 1.17 | 3.923 | 1.14 | 3.883 | 1.05 | 3.427 | 1.01 |
| 4000×1000 | 16 | 2.117 | 1.07 | 2.012 | 1.03 | 2.029 | 1.03 | **1.918** | 0.98 |
| 4000×4000 | 8 | **114.5** | 0.98 | **114.2** | 0.99 | **120.3** | 0.94 | **108.2** | 0.93 |
| 4000×4000 | 16 | 58.53 | 1.04 | 57.38 | 1.03 | **54.69** | 0.97 | **54.31** | 0.97 |
| 5000×100 | 8 | **0.050** | 0.90 | **0.046** | 0.83 | **0.049** | 0.86 | **0.045** | 0.80 |
| 5000×1000 | 16 | 2.538 | 1.15 | 2.427 | 1.10 | 2.335 | 1.05 | 2.239 | 1.01 |
| 5000×5000 | 16 | 112.8 | 1.06 | 112.5 | 1.06 | **105.7** | 0.99 | **106.0** | 0.97 |
| 8000×1000 | 16 | 3.776 | 1.26 | 3.660 | 1.24 | 3.762 | 1.15 | 3.241 | 1.08 |
| 8000×8000 | 16 | 638.2 | 1.05 | 635.9 | 1.00 | **620.9** | 0.99 | 670.9 | 1.06 |
| 10000×1000 | 16 | 5.613 | 1.46 | 5.450 | 1.42 | 4.588 | 1.18 | 4.662 | 1.21 |
| 10000×10000 | 16 | **1231** | 0.95 | **1226** | 0.97 | **1129** | 0.87 | **1216** | 0.96 |

$t_5$ — the parallel SVD with a block one-sided bidiagonalization algorithm. It is a simplified block version of the one-sided bidiagonalization. Only the application of the Householder reflectors is aggregated, but the BLAS 2.5 approach is not used since it introduces one matrix-vector product more per iteration.

$t_6$ — the parallel SVD with a modified block one-sided bidiagonalization algorithm. It is the blocked version with the modification of Campos et al.

As we can see from Table 6.3, the situation on the Sarek cluster is a little bit different from the situation described in subsection 6.1. The Sarek cluster has very fast interprocessor communication and Algorithm 5.1 is not so favorable as it was on the cluster of computers in the Advanced Computing Laboratory in the case when $m > n$. The worst result was obtained for matrix dimensions $10000 \times 1000$ and 16 processors when the execution time of the parallel SVD with Algorithm 5.1 was 1.46 times longer than the execution time of `psgesvd()`. This was the reason for developing three additional implementations of the parallel stable one-sided bidiagonalization routines, which are explained in the description of Table 6.3. The modification of Campos and his coauthors proposes reduction in communication events. This is achieved by simultaneous execution of the operations in steps **(6)** and **(12)** of Algorithm 5.1, since each of these operations presents a communication event. Thus instead of two communication events, only one is performed per iteration. The work of Campos et al. was developed independently of the work presented in this paper. The block parallel implementation is similar to Algorithm 3.1, except that BLAS 2.5 operations are not implemented. We can conclude that the presented variants of Algorithm 3.1 did gain some speedup and that block and modified block algorithms are faster than the

ScaLAPACK routine in most cases. For some matrix dimensions these two variants are up to 21.5% faster. On the other hand, the matrix dimension 10000 × 1000 for 16 processors remains critical. We achieved a reduction in slowdown from 45.53% to 18.25%, but the ScaLAPACK routine is still faster in this case. There is still an open question of whether we can make our parallel implementation of the one-sided bidiagonalization more efficient on the clusters with a fast network. This will be the subject of our future work.

**7. Conclusion.** Ralha's one-sided bidiagonalization [29] performs fewer operations than the standard Golub–Kahan bidiagonalization [20], but it is numerically unstable. Barlow's modification [3] of Ralha's algorithm avoided this problem, so the new algorithm became numerically backward stable for computing the matrices $\Sigma$ and $V$, with the same operation count. On modern computers, a smaller operation count does not necessarily lead to reduced execution times, due to the time spent on communication between different levels of memory. This is the reason why we have developed block and parallel versions of the new stable one-sided bidiagonalization algorithm. The block version optimizes the usage of faster memory, without sacrificing numerical backward stability. As numerical tests demonstrate, the SVD algorithm with the block one-sided bidiagonalization is faster than the corresponding LAPACK routine. The stable one-sided bidiagonalization is more suitable for parallelization than the corresponding ScaLAPACK routine. The tests established improvement of the parallel one-sided bidiagonalization when compared with the ScaLAPACK routine in most cases. In the best case our algorithm is 68.28% faster than the ScaLAPACK routine on a cluster with slow interprocessor communication and 21.5% faster on a cluster with fast interprocessor communication. From the numerical point of view it is equivalent to the original stable one-sided algorithm proposed in [3] or its block version and thus numerically stable in the same way. In the future we will work on the more efficient parallel version of the stable one-sided bidiagonalization.

**Acknowledgments.** We thank Zlatko Drmač and Daniel Kressner for carefully reading the paper and making many helpful suggestions. Rui Ralha and Carlos Campos are developing one-sided bidiagonalization software, and we thank them for their support. We also thank both referees for helping us to make this paper better. This research was conducted using the resources of the High Performance Computing Center North (HPC2N).

## REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, and D. C. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[2] J. L. Barlow, *More accurate bidiagonal reduction for computing the singular value decomposition*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 761–798.

[3] J. L. Barlow, N. Bosner, and Z. Drmač, *A new stable bidiagonal reduction algorithm*, Linear Algebra Appl., 397 (2005), pp. 35–84.

[4] C. Bischof and C. Van Loan, *The WY representation for products of Householder matrices*, SIAM J. Sci. Comput., 8 (1987), pp. s2–s13.

[5] Å . Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190.

[6] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. J. Dongarra, S. J. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*, SIAM, Philadelphia, 1997.

[7] N. Bosner, *Fast methods for large scale singular value decomposition*, Ph.D. thesis, Department of Mathematics, University of Zagreb, 2006, http://www.math.hr/∼nela/eng.html.

[8] N. Bosner and Z. Drmač, *On accuracy properties of one-sided bidiagonalization algorithm and its applications*, in Proceedings of the Conference on Applied Mathematics and Scientific Computing, Z. Drmač, M. Marušić, and Z. Tutek, eds., Springer, Dordrecht, 2005, pp. 141–150.

[9] C. Campos, D. Guerrero López, V. Hernandez, and R. Ralha, *Parallel bidiagonalization of a dense matrix*, SIAM J. Matrix Anal. Appl., to appear.

[10] T. F. Chan, *An improved algorithm for computing the singular value decomposition*, ACM Trans. Math. Software, 8 (1982), pp. 72–83.

[11] J. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[12] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[13] J. J. Dongarra, J. J. DuCroz, S. J. Hammarling, and R. J. Hansen, *An extended set of Fortran basic linear algebra subroutines*, ACM Trans. Math. Software, 14 (1988), pp. 1–17.

[14] J. J. Dongarra, J. Du Croz, S. J. Hammarling, and I. S. Duff, *A set of level* 3 *basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.

[15] J. J. Dongarra, J. Du Croz, S. J. Hammarling, and I. S. Duff, *Algorithm* 679; *A set of level* 3 *basic linear algebra subprogram: Model implementation and test programs*, ACM Trans. Math. Software, 16 (1990), pp. 18–28.

[16] J. J. Dongarra, S. J. Hammarling, and D. C. Sorensen, *Block reduction of matrices to condensed forms for eigenvalue computations*, J. Comput. Appl. Math., 27 (1989), pp. 215–227.

[17] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm:* I, SIAM J. Matrix Anal. Appl., to appear.

[18] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm:* II, SIAM J. Matrix Anal. Appl., to appear.

[19] K. V. Fernando and B. N. Parlett, *Accurate singular values and differential* qd *algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[20] G. Golub and W. Kahan, *Calculating the singular values and pseudoinverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[21] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, Numer. Math., 14 (1970), pp. 403–420.

[22] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, 2nd ed., Scientific and Engineering Computation Series, MIT Press, Cambridge, 1999.

[23] B. Grosser and B. Lang, *Efficient parallel reduction to bidiagonal form*, Parallel Comput., 25 (1999), pp. 969–986.

[24] B. Grosser and B. Lang, *An* $\mathcal{O}(n^2)$ *algorithm for the bidiagonal SVD*, Linear Algebra Appl., 358 (2003), pp. 45–70.

[25] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[26] G. Howell, C. Fulton, J. Demmel, S. J. Hammarling, and K. Marmol, *Cache efficient bidiagonalization using BLAS* 2.5 *operators*, Lapack Working Note 174, 2006.

[27] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[28] R. Ralha, *A new algorithm for singular value decompositions*, in Proceedings of the 3rd Euromicro Workshop on Parallel and Distributed Processing, IEEE Press, Piscataway, NJ, 1994, pp. 240–244.

[29] R. Ralha, *One-sided reduction to bidiagonal form*, Linear Algebra Appl., 358 (2003), pp. 219–238.

[30] R. Ralha and A. Mackiewicz, *An efficient algorithm for the computation of singular values*, in Proceedings of the III International Congress on Numerical Methods in Engineering, M. Doblaré, J. M. Correas, L. Gaverte, and M. Pastor, eds., SEMNI (Sociedad Española de Métodos Numéricos en Ingenieria), 1996, pp. 1371–1380.

[31] H. D. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.

[32] P. R. Willems, B. Lang, and C. Vömel, *Computing the bidiagonal SVD using multiple relatively robust representations*, Special Issue on Accurate Solution of Eigenvalue Problems, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 907–926.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 3, pp. 954–971

# A CHEBYSHEV–DAVIDSON ALGORITHM FOR LARGE SYMMETRIC EIGENPROBLEMS*

YUNKAI ZHOU† AND YOUSEF SAAD‡

**Abstract.** A polynomial filtered Davidson-type algorithm is proposed for symmetric eigenproblems, in which the correction-equation of the Davidson approach is replaced by a polynomial filtering step. The new approach has better global convergence and robustness properties when compared with standard Davidson-type methods. The typical filter used in this paper is based on Chebyshev polynomials. The goal of the polynomial filter is to amplify components of the desired eigenvectors in the subspace, which has the effect of reducing both the number of steps required for convergence and the cost in orthogonalizations and restarts. Numerical results are presented to show the effectiveness of the proposed approach.

**Key words.** polynomial filter, Davidson-type method, global convergence, Krylov subspace, correction-equation, eigenproblem

**AMS subject classifications.** 15A18, 15A23, 15A90, 65F15, 65F25, 65F50

**DOI.** 10.1137/050630404

## 1. Introduction.

We consider a Davidson-type method for the standard eigenvalue problem

$$Au = \lambda u, \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric, $n$ is large, and a large number of eigenpairs need to be computed. We assume throughout that the eigenvalues wanted are the smallest ones. There is a growing need for solving this type of problem efficiently. To cite just one example, symmetric eigenvalue problems usually constitute the most time-consuming part of electronic structure calculations [18, 12, 6].

The original Davidson method [11] was initially designed for diagonally dominant matrices, which for eigenvalue problems means matrices whose off-diagonal elements are small compared with the changes in magnitude between diagonal elements [19]. The Davidson approach sacrifices the attractive Krylov subspace structure, at the cost of having to compute eigenpairs and associated residual vectors of a projection matrix at each (outer) iteration. The trade-off is that the Davidson approach can augment the subspace by a new vector potentially much better than the one based on a strict Krylov subspace structure.

The "augmentation vector" added to the subspace at each step usually results from solving a correction-equation. The efficiency of the standard Davidson-type methods depends on the quality of the correction-equation used. Efficient (preconditioned) linear equation solvers are often utilized to solve the correction-equations. The original Davidson method uses the correction-equation $(\mathrm{diag}(A) - \mu I)t = -r$, where

---

*Received by the editors May 2, 2005; accepted for publication (in revised form) by D. Calvetti March 28, 2007; published electronically October 5, 2007. This work was supported by the U.S. Department of Energy under contract DE-FG02-03ER25585, by NSF grants ITR-0428774 and CMMI-0727194, and by the Minnesota Supercomputing Institute.

http://www.siam.org/journals/simax/29-3/63040.html

†Department of Mathematics, Southern Methodist University, Dallas, TX 75275 (yzhou@smu.edu).

‡Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 (saad@cs.umn.edu).

$r = Ax - \mu x$ is the residual vector corresponding to a Ritz pair $(\mu, x)$, and $t$ is the augmentation vector to be computed. In [19] better approximations of $A$ are used to replace the diagonal of $A$ in the correction equation. However, it was noted in [19] that using the "exact preconditioner" leads to stagnation, since $t = (A - \mu I)^{-1} r = -x$ cannot augment the subspace. This led to the development of the more efficient Jacobi–Davidson (JD) algorithm [32, 14, 31]. Work in the literature has shown that the JD can be competitive with efficient Krylov subspace methods such as those in [33, 17, 36, 42, 45]. In [41, 43] other correction-equations for Davidson-type methods are derived.

The JD method can be related to the Newton method or the approximate Rayleigh-quotient iteration (RQI). As such, it has been observed that the method can be rather slow if the starting vector is far away from the desired eigenvector. We note that even though RQI is globally convergent for symmetric eigenproblems (see [22], [23, p. 81]), it may converge to an unwanted eigenpair. The global convergence can be slow when the approximate RQI is used in a subspace method and the method is required to converge to wanted eigenpairs. Global acceleration schemes for JD have been studied. For example, in [5] a nonlinearized JD correction-equation is proposed; however, the preconditioning may be difficult to apply for the nonlinearized correction-equation, and the approach can become much more expensive than the JD method when the number of desired eigenpairs is large. Another approach to achieving better global convergence, as suggested in [19, 14], is to apply an Arnoldi or Lanczos method to get a good initial vector and then apply the JD algorithm.

In this paper, we explore a different Davidson-type approach called the Chebyshev–Davidson method. There is no need to form or solve any correction-equations within this approach; instead, intervalwise filtering based on Chebyshev polynomials is utilized.

The Chebyshev–Davidson approach is very suitable for problems where solving (preconditioned) equations is expensive, e.g., when good preconditioners for correction-equations are either unknown or too expensive to construct.

Details of the Chebyshev–Davidson method are presented in sections 3–4. Comparisons with existing representative eigenvalue algorithms are in section 6. The Chebyshev–Davidson method (written in Fortran95) has been applied to solve a class of highly challenging problems with dimension over a few millions, where more than ten thousand eigenpairs need to be computed.

## 2. Advantages of polynomial filtering.

The global convergence of a Davidson-type method can be improved in a natural and systematic way via polynomial filtering. We first make the following three observations. The first is on the well-known polynomial filtering argument: For a symmetric matrix $A$ with the eigendecomposition $A = Q\Lambda Q^T$, any polynomial $\psi(s) : \mathbb{R} \to \mathbb{R}$ satisfies

$$\psi(A)v = Q\psi(\Lambda)Q^T v \quad \forall v \in \mathbb{R}^n. \tag{2.1}$$

The second observation is on the fast local convergence of JD. It is shown in [43] that the locally fast convergence of JD is mainly caused by the retention of the approximate RQI direction in the basis of the projection subspace. Assume throughout that $(\mu, x)$ denotes the current Ritz pair that best approximates a wanted eigenvalue, and the Ritz vector $x$ is of unit length, and let $r = Ax - \mu x$ denote the residual. It was observed in [43] that the JD correction equation,

$$\text{Solve for } t \perp x \text{ from } \ (I - xx^T)(A - \mu I)(I - xx^T)t = r, \tag{2.2}$$

can be simplified to

$$\text{Solve for } t \text{ from } \ (I - xx^T)(A - \mu I)t = r. \tag{2.3}$$

The right projection by $(I - xx^T)$ and the final orthogonality constraint $t \perp x$ can be omitted. It is the approximate RQI direction, which is an approximation to $(A - \mu I)^{-1}x$, that leads to the success of the JD approach. The left projector $(I - xx^T)$ in (2.2) is crucial in retaining the important approximate RQI direction in the JD direction (solution $t$ of (2.2)). This can be readily seen by writing (2.2) or (2.3) as

$$(A - \mu I)t = r + x\ \alpha, \tag{2.4}$$

where $\alpha$ is a nonzero scalar. The left projector also improves the conditioning of (2.2) and (2.3) on the $x^{\perp}$ subspace—the subspace in which a vector to augment the current projection subspace is sought, but this property is irrelevant for this paper.

Note that the exact RQI direction is $(A-\mu I)^{-1}x$, which is the current Ritz vector $x$ filtered by the rational polynomial $\varphi(s) = \frac{1}{s-\mu}$. This polynomial significantly magnifies the direction of a possibly wanted eigenvector corresponding to the Ritz value $\mu$ (the current best approximation to a wanted eigenvalue of $A$).

The third observation is that one can improve global convergence by choosing a polynomial $\psi(s)$ which magnifies not only the direction corresponding to one single point, but also directions corresponding to an interval containing wanted eigenvalues, and at the same time dampens unwanted eigenvalues. With polynomial filtering, it is unlikely that wanted eigenvalues will be missed, because when the whole interval containing wanted eigenvalues is magnified, so is each wanted eigenvalue in the interval. In contrast, standard Davidson-type methods may miss some wanted eigenvalues. This is because correction-equations often resemble certain shift-invert formulations, and the shift chosen at some step may approximate larger eigenvalues before all the wanted smaller eigenvalues are computed. Chebyshev filtering offers an alternative which can improve global convergence as well as robustness (in the sense that wanted eigenvalues are not missed) of Davidson-type methods.

Note that polynomial filters have long been exploited to accelerate Arnoldi/Lanczos algorithms; see, e.g., [26, 33]. Here we consider a natural application of Chebyshev polynomials within a Davidson-type (non-Krylov) framework. This approach combines the acceleration power of the Chebyshev filtering technique and the flexibility and robustness of the Davidson approach.

To further explain the third observation, we suppose that the eigenvalues of $A$ are ordered as $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, and that the wanted eigenvalues are located in $[\lambda_1, \lambda_k]$. If $\psi(s)$ is chosen to approximate the step function

$$\phi(s) = \begin{cases} 1, & \lambda_1 \le s \le \lambda_k, \\ 0, & \lambda_k < s \le \lambda_n, \end{cases} \tag{2.5}$$

then (2.1) shows that $\psi(A)v \approx \sum_{i=1}^{k} \alpha_i q_i$, where $q_i$ is the $i$th column of $Q$ and $\alpha_i = q_i^T v$. That is, $\psi(A)v$ is contained in the subspace spanned by the wanted eigenvectors $Q(:, 1 : k)$. If this $\psi(A)v$ is augmented into the basis, convergence to the wanted eigenvectors is expected to be much faster than augmenting the basis by a vector closer to unwanted eigenvectors. This claim can be verified by explicitly computing $Q$ and using $\sum_{i=1}^{k} \alpha_i q_i$ $(\alpha_i = q_i^T x$, where $x$ denotes the current Ritz vector at each iteration) as the augmentation vector in a Davidson-type method. This is equivalent to using a filter that exactly approximates (2.5). Note that this filter leads to no gap among wanted eigenvalues, but in this ideal setting it can still lead to fast convergence in a Davidson-type method. However, a low degree polynomial cannot approximate (2.5) well. In real computations, a filter that can introduce more favorable gaps for the wanted eigenvalues is far better than others that introduce no gap.

For a subspace method applied to (1.1), the essence in obtaining fast convergence is in augmenting the subspace by vectors close to the wanted invariant subspace of $A$. Therefore, a convergence acceleration scheme should construct a suitable filter $\psi$ so that the vector $\psi(A)v$ used for augmentation is contained in the wanted eigensubspace. By this filtering we obtain better global convergence.

According to observations just made, it is essential to filter the current Ritz vector $x$, not the residual vector $r$. Note that at each iteration of a Davidson-type method, $r$ is orthogonal to the projection basis, and this basis is used to approximate the wanted eigenvectors; hence $r$ can become orthogonal to the wanted eigenvectors during the iteration. The residual vector $r$ is not suitable for the filtering because, when $Q(:,1:k)^T r \approx 0$, $\psi(A)r = Q\psi(\Lambda)Q^T r$ is approximately inside the subspace spanned by unwanted eigenvectors.

We also mention that our own experiments, together with those in [13], show that the preconditioned Davidson method based on equation $(A - \mu I)t = r$ can be inefficient because of the higher possibility of stagnation if this equation is solved more accurately. An efficient correction-equation essentially should retain the approximate RQI direction in its solution. Thus, the JD method is equivalent to (2.4), but the $x$ term in the right-hand side of (2.4) may be more important than the $r$ term. However, in the case of rather inaccurate solves with a fixed preconditioner, [1] shows that $(A - \mu I)t = r$ can have better performance than other correction equations.

**3. Chebyshev polynomial filter.** The observations in section 2 suggest that polynomials which significantly magnify the lowest end of a wanted interval and dampens unwanted intervals at the same time can be used as a filter to improve global convergence. The well-known Chebyshev polynomials are a natural choice for this task. Using Chebyshev polynomials to accelerate symmetric eigenvalue computations dates back to [24, 25]. A nice discussion on Chebyshev accelerated subspace iteration can be found in [23, pp. 329–330], where it is mentioned that the Lanczos method is usually better than the Chebyshev accelerated (fixed dimension) subspace iteration algorithm. Here we integrate Chebyshev filtering into a varying dimension Davidson-type algorithm.

With Chebyshev acceleration, the subspace used in a Davidson-type method can be of much smaller dimension than that is required by a Lanczos-type method for good efficiency. Therefore the filtering approach leads to substantial savings in (re-) orthogonalization costs.

Recall that the real Chebyshev polynomials of the first kind are defined by (see, e.g., [23, p. 371], [27, p. 142])

$$C_k(t) = \begin{cases} \cos(k\, cos^{-1}(t)), & -1 \le t \le 1, \\ \cosh(k\, cosh^{-1}(t)), & |t| > 1. \end{cases}$$

Note that $C_0(t) = 1, C_1(t) = t$. Recall also the important three-term recurrence,

$$C_{k+1}(t) = 2tC_k(t) - C_{k-1}(t), \qquad t \in \mathbb{R}. \tag{3.1}$$

A remarkable property of the Chebyshev polynomial is its rapid growth outside the interval $[-1, 1]$. This property is illustrated in Figure 3.1. Here we plot only the

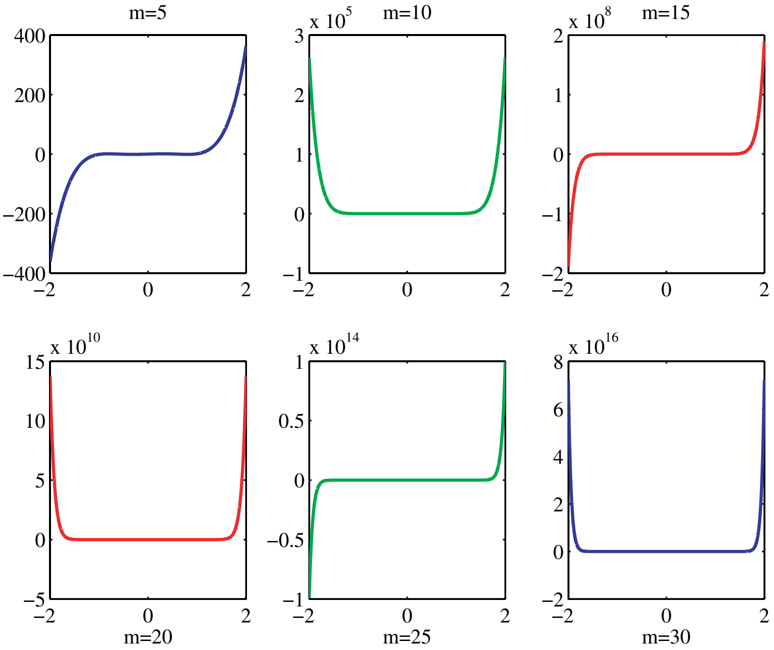


FIG. 3.1. *Rapid increase outside* $[-1, 1]$ *of Chebyshev polynomial of degree* $m$.

polynomial on the $[-2, 2]$ interval, but note that the farther away we are from $[-1, 1]$, the larger the magnitude of $C_k(t)$. Suppose that the spectrum of $A$ is contained in $[a_0, b]$ and we want to dampen the interval $[a, b]$ for $a > a_0$; then we need only to map $[a, b]$ into $[-1, 1]$ by an affine mapping. This mapping will map the wanted lower end of the spectrum, i.e., the eigenvalues closer to $a_0$, farther away from $[-1, 1]$ than the ones closer to $a$. Applying the three-term Chebyshev recurrence will then magnify eigenvalues near $a_0$ and dampen eigenvalues in $[a, b]$, which is the desired filtering.

In practice, we need the lower bound $a$ of the unwanted interval, which is easy to approximate during each iteration in a Davidson-type method. The upper bound $b$ of the eigenvalues of $A$ can by obtained by Gerschgorin's theorem. It can also be estimated by an upper-bound-estimator (Algorithm 4.3 in [47]), which applies a few steps of Lanczos iteration with a final safeguard step.

The Chebyshev iteration, which dampens values in $[a, b]$ while magnifying values in the interval to the left of $[a, b]$, is presented in Algorithm 3.1 below. Here we follow the formula derived in [26], [27, p. 223] for the complex Chebyshev iteration and adapt it to the real case. The iteration of the algorithm is equivalent to computing

$$y = p_m(A)x, \qquad \text{where} \quad p_m(t) = C_m\left(\frac{t-c}{e}\right). \tag{3.2}$$

As defined in the algorithm, $c$ is the center of the interval $[a, b]$ and $e$ its half-width; both depend on the bounds used.

In Algorithm 3.1, the $\sigma$'s are used for scaling purposes; the $a_0$ is a crude approximation of the smallest eigenvalues of $A$. The following discusses certain details of scaling. The three-term recurrence using $p_m(A)$ yields the iteration

$$x_{j+1} = \frac{2}{e}(A - cI)x_j - x_{j-1}, \quad j = 1, 2, \ldots, m-1,$$

with $x_0$ given and $x_1 = (A - cI)x_0$. This is equivalent to a power iteration of the form

$$\begin{pmatrix} x_{j+1} \\ x_j \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{2}{e}(A - cI) & -I \\ I & 0 \end{pmatrix}}_{\mathcal{B}} \begin{pmatrix} x_j \\ x_{j-1} \end{pmatrix} .$$

A little analysis would show that all the eigenvalues of the nonsymmetric matrix $\mathcal{B}$ are complex and of modulus one, except that those corresponding to eigenvalues of $A$ that are less than $a$ are mapped to real eigenvalues larger than one in magnitude. Therefore, as for the standard power method, a scaling at each step is recommended. The simplest strategy, discussed in [27], is to consider the scaled sequence

$$\tilde{x}_j = \frac{C_j[\frac{2}{e}(A - cI)]}{C_j[\frac{2}{e}(a_0 - cI)]}\, x_0,$$

where $\rho_j = C_j[\frac{2}{e}(a_0 - cI)]$ is the scaling factor. This requires $a_0$, but since it is only used for scaling, a rough estimate of $a_0$ is sufficient. For the first Chebyshev–Davidson iteration, we can use a value $a_0 \leq a$; for the latter Chebyshev–Davidson steps, the smallest Ritz value from the previous step can be used. The vector sequence is not computed as shown above, because $\rho_j$ itself can be large and this would defeat the purpose of scaling. Instead, each $\tilde{x}_{j+1}$ is updated using the scaled vectors $\tilde{x}_j$ and $\tilde{x}_{j-1}$. This is discussed in [27], and Algorithm 3.1 implements this scaling (the tildes and vector subscripts are omitted).

```
ALGORITHM 3.1. [illegible] Chebyshev_filter $x, m, a, b, a_0$
[illegible] $x$ [illegible] $m$ [illegible] $[a, b]$
    $e = (b - a)/2;$   $c = (b + a)/2;$
    $\sigma = e/(a_0 - c);$   $\sigma_1 = \sigma;$
    $y = (Ax - cx)\sigma_1/e;$
    [illegible] $i = 2 : m$
        $\sigma_{new} = \frac{1}{(2/\sigma_1 - \sigma)};$
        $y_{new} = 2(Ay - cy)\sigma_{new}/e - \sigma\sigma_{new}x;$
        $x = y;$   $y = y_{new};$   $\sigma = \sigma_{new};$
    [illegible]
```

Note that the filter is intervalwise; hence no shifts such as Chebyshev zeros or Leja points are required. This is to be contrasted with pointwise filtering methods, e.g., [33, 17, 2, 3].

Clearly, polynomials other than Chebyshev can be used for filtering. Several polynomials are discussed in [34], with emphasis on approximating rational functions of the form $\varphi(s) = 1/(s - \mu)$. In contrast, we do not approximate the (shift-inverted) rational functions, but require polynomials to have the desired intervalwise filtering property, i.e., dampen an interval and significantly magnify other intervals. We choose Chebyshev polynomials because of their desirable filtering properties and ease of implementation. Note that the Chebyshev filtering used in [34] is different from Algorithm 3.1, since the former requires an additional parameter $\Delta$, which is not straightforward to specify. Another difference is that in [34] Chebyshev filtering is used in a Lanczos-type algorithm, while here we integrate Chebyshev filtering into a Davidson-type framework. The Rayleigh–Ritz step in a Davidson-type method readily provides the necessary bounds for constructing efficient Chebyshev filters. The resulting Chebyshev–Davidson method compares favorably with other methods, as shown in section 6.

Algorithm 3.1 requires no inner products, and this is another appealing feature of Chebyshev acceleration, since inner products incur a global reduction which requires additional communication costs in a parallel computing context.

## 4. Chebyshev polynomial accelerated Davidson method.

The pseudocode for the Chebyshev–Davidson method is presented in Algorithm 4.1 below. This code is very different from other Davidson-type methods in the literature (e.g., [4]). We use a natural but useful indexing scheme. The deflation of converged eigenvectors is handled by indexing the columns of the projection basis $V$. No extra storage for the converged eigenvectors is necessary. Moreover, restarting is simplified (as seen in step 8f) by the indexing. The implementation does not require extra basis updates or memory copies during the restart, since the updates in step 8g need to be performed even when restart is not necessary. We note that putting restart at step 8f is better than putting it at the end of the outer loop, because it saves operations in step 8g when restarting is necessary.

We make a few comments on Algorithm 4.1. Comments (v)–(vii) are related to the robust implementation of any Davidson-type methods.

(i) It is important that the bound `upperb` bounds all eigenvalues of $A$ from above. Otherwise the interval containing largest eigenvalues may also be magnified through filtering, and this can drastically slow convergence or even lead to wrong convergence. One inexpensive way for the bound estimation at step 5 is `upperb` $= \left|\left|A\right|\right|_1$; if $A$ is available only through a matrix-vector-product subroutine, then we can apply the upper-bound-estimator, Algorithm 4.1 in [47], to get an upper bound.

(ii) The choice of the lower bound for the unwanted interval at each iteration is one of the most critical ingredients of the method. The Chebyshev–Davidson method allows quite flexible choices for this lower bound, without any extra computations. Numerical results show that the choice at step 8l is remarkably efficient. Other choices, such as the maximum of the current Ritz values, can also be used as `lowerb`.

(iii) For the orthogonalization step 8b, we use the iterated Gram–Schmidt algorithm, often referred to as the DGKS method [10].

(iv) The refinement at step 8g is performed at each step. One can avoid this step until some eigenpair converges. But according to [23, p. 325], this refinement is necessary in order to have faster convergence for the eigenvectors.

(v) The swap at step 8j may be performed by the following pseudocode:

```
set $v_{tmp} = V(:, k_c)$;
For $(i = k_c - 1 : -1 : 1)$ Do
    If $(\mu \geq eval(i))$, exit the For loop; End If
    set $eval(i+1) = eval(i)$, $eval(i) = \mu$; set $V(:, i+1) = V(:, i)$, $V(:, i) = v_{tmp}$;
End For.
```

(Note that unnecessary memory copies in the above can be avoided with some more involved programming.)

(vi) The *noswap* flag at steps 8i–k is used to improve robustness. This flag decreases the possibility of counting converged unwanted eigenvalues as wanted ones.

(vii) At step 8j, a convergence test is performed only on the first Ritz pair among the $k_{sub} - k_c$ Ritz pairs available at each iteration. A simple loop can be added to check the convergence of more than one Ritz pair. We note that for almost all Davidson-type subspace methods, if all the $k_{sub} - k_c$ Ritz pairs are checked for convergence at each iteration step and no swap procedure is included, then there is a high possibility of missing wanted eigenvalues.

(viii) Algorithm 4.1 essentially contains a framework for Davidson-type methods based on filtering. The Chebyshev filter at step 8a can be replaced by other suitable filters. However, we mention that among the filters tried, including a least square polynomial and a different implementation of Chebyshev polynomials in [34], the filter as implemented in Algorithm 3.1 has the best numerical behavior.

**5. Analysis.** The analysis given here serves to give a preliminary understanding of the convergence for Algorithm 4.1. Therefore several simplifications of the algorithm are made.

Assume that the eigenvalues of $A$ are $\lambda_1 < \lambda_2 \le \cdots \le \lambda_n$, and denote the associated unit eigenvectors by $q_1, \dots, q_n$. According to (3.2) and the fact that the

```
ALGORITHM 4.1. Chebyshev–Davidson method.
[illegible] $k_{want}$ [illegible]
Input: $x$– [illegible] $m$– [illegible] $k_{keep}$– # [illegible]
       [illegible] $dim_{max}$– [illegible] $\tau$– [illegible]
Output: [illegible] $eval(1 : k_c)$ [illegible]
       [illegible] $V(:, 1 : k_c)$ [illegible] $k_c$ [illegible] # [illegible]
1. [illegible] $x$ $V = [x]$.
2. [illegible] $W = [Ax], H = [\mu]$ [illegible] $\mu = x^T w$.
3. [illegible] $r = W(:, 1) - \mu x$
4. [illegible] $||r|| <= \tau$ [illegible] $eval(1) = \mu$ $k_c = 1$ $H = [\,]$. [illegible] $k_c = 0$
5. [illegible] upperb [illegible]
6. [illegible] lowerb $=$ (upperb $+ \mu)/2$ $a_0 =$ lowerb
7. [illegible] $k_{sub} = 1$ $k_{sub}$ [illegible]
8. [illegible] Do while $iter \le iter_{max}$
   a. [illegible]
      $[t]$ [illegible] Chebyshev_filter $x, m,$ lowerb, upperb $a_0$
   b. [illegible] $t$ [illegible] $V(:, 1 : k_{sub})$ [illegible] $V(:, k_{sub} + 1)$.
      [illegible] $k_{sub} \leftarrow k_{sub} + 1$. [illegible] $k_{old} \leftarrow k_{sub}$.
   c. [illegible] $W(:, k_{sub}) = AV(:, k_{sub})$
   d. [illegible] $H$
      $H(1 : k_{sub} - k_c, k_{sub} - k_c) = V(:, k_c + 1 : k_{sub})^T W(:, k_{sub})$
   e. [illegible] $H$ $HY = YD$
      [illegible] diag($D$) [illegible] $\mu = D(1, 1)$
   f. [illegible] $k_{sub} \ge dim_{max})$ [illegible] $k_{sub} = k_c + k_{keep}$
   g. [illegible] $V(:, k_c + 1 : k_{sub}) \leftarrow V(:, k_c + 1 : k_{old})Y(:, 1 : k_{sub} - k_c)$.
      [illegible] $W$ $W(:, k_c + 1 : k_{sub}) \leftarrow W(:, k_c + 1 : k_{old})Y(:, 1 : k_{sub} - k_c)$
   h. [illegible] $r = W(:, k_c + 1) - \mu V(:, k_c + 1)$.
   i. [illegible] $noswap = 0$ $iter \leftarrow iter + 1$
   j. [illegible] $||r|| <= \tau \max(\text{diag}(D))$ [illegible] $k_c = k_c + 1$
      [illegible] $eval(k_c) = \mu$. [illegible] (v)
      [illegible]
      [illegible] $noswap = 1$ [illegible]
   k. [illegible] $k_c \ge k_{want}$ [illegible] $noswap == 0$ Return $eval(1 : k_c)$ [illegible] $V(:, 1 : k_c)$ [illegible]
      [illegible] Exit
   l. [illegible] lowerb $= median(\text{diag}(D))$.
      [illegible] $a_0 > \min(\text{diag}(D))$ [illegible] $a_0 \leftarrow \min(\text{diag}(D))$.
   m. [illegible] $x = V(:, k_c + 1)$
   n. [illegible] $H$ $H = D(k_c + 1 : k_{sub}, k_c + 1 : k_{sub})$
```

interval of the eigenvalues to be dampened at each step is adaptively changing, we see that the matrix applied at the $j$th step is

$$p_m^{(j)}(A) = C_m^{(j)}((A - c_jI)/e_j). \tag{5.1}$$

The first simplification assumes that the interval of the eigenvalues to be dampened at each filtering step of Algorithm 4.1 is fixed. That is, the matrix involved is fixed as $p_m(A) \equiv C_m((A - cI)/e)$. The second simplification assumes that no restart is used in the algorithm.

We further assume that Algorithm 4.1 is a [illegible] version. That is, in step 8b we keep $k_{sub} \equiv 1$ and set $V(:,1) = t/||t||$, and in step 8m we set $x = V(:,1)$. Then the algorithm becomes a standard power method with the matrix $p_m(A)$. As a result, the convergence will be governed by the ratio of the two dominant eigenvalues. Note that the interval $[a, b]$ of the eigenvalues to be dampened satisfies $\lambda_1 < a$. The (unique) dominant eigenvalue of the matrix $p_m(A)$ is $C_m((\lambda_1 - c)/e)$. So, in the one-dimensional version of the algorithm, $V(:,1)$ converges to $q_1$ with the convergence factor

$$\rho = \frac{\max_{j>1} |C_m((\lambda_j - c)/e)|}{|C_m((\lambda_1 - c)/e)|} < 1.$$

Consider now the situation in which $k_{sub}$ can be increased. The simplified method turns out to have a simple Krylov interpretation. Assume that we perform two steps of the algorithm, i.e., that the dimension of the subspace is two. The first vector of the basis is $p_m(A)x$. The second is obtained as $p_m(A)x_1$, where $x_1$ is an approximate eigenvector from the one-dimensional space spanned by the first vector, which is simply a multiple of $p_m(A)x$. The subspace used in this case is

$$K_2 = \text{span}\{p_m(A)x, p_m(A)x_1\} = \text{span}\{p_m(A)x, p_m^2(A)x\} \ ,$$

which is the Krylov subspace of dimension two usually denoted by $K_2(p_m(A), x)$. Consider now the third step. The process will inject to the subspace a vector of the form

$$p_m(A)x_2 \quad \text{with} \quad x_2 \in K_2 \ .$$

The vector $x_2$ is a Ritz eigenvector computed from projecting $A$ onto the subspace $K_2$, and it is a linear combination of vectors from $K_2$, so we can write $x_2 = \alpha_1 p_m(A)x + \alpha_2 p_m^2(A)x$. The new subspace $K_3$ is again a Krylov subspace. Indeed,

$$K_3 = \text{span}\{p_m(A)x, p_m^2(A)x, p_m(A)x_2\} \equiv \text{span}\{p_m(A)x, p_m^2(A)x, p_m^3(A)x\} \ .$$

The result can be easily extended to an arbitrary step $j$ for the simplified method.

PROPOSITION 5.1. [illegible] $j$ [illegible] 4.1 [illegible] $A$ [illegible]

$$K_j\left(p_m(A), x\right).$$

In particular, this means that if one generated an orthogonal basis $V_j$ of the Krylov subspace $K_j(p_m(A), x)$ and computed the eigenvalues of $V_j^TAV_j$, these eigenvalues would be identical with those of the simplified Algorithm 4.1. This is not quite a

Krylov subspace method, because the projection uses $A$ instead of the transformed matrix $p_m(A)$. However, this simple result permits one to analyze the simplified algorithm in a complete way by considering eigenvectors. Indeed, eigenvectors of $A$ and $p_m(A)$ are identical, and there are results which establish upper bounds for the angle between the exact eigenvector and the Krylov subspace. This will be omitted, and the reader is referred to [23] for details.

Although the simplified algorithm can be viewed from the angle of Krylov subspaces, Algorithm 4.1 is not a Krylov method. There are a number of distinguishing features, related to implementation and other practical aspects. In fact, Proposition 5.1 assumes that the polynomial is fixed, but the actual Chebyshev–Davidson method adapts the filters by dynamically adjusting the bounds of the interval of eigenvalues to be dampened at each iteration. This in practice leads to significantly more efficient filters. The following presents a conservative analysis: Taking (5.1) into account, we see that

$$K_2 = \operatorname{span}\{p_m^{(1)}(A)x, p_m^{(2)}(A)p_m^{(1)}(A)x\},$$

$$K_3 = \operatorname{span}\{p_m^{(1)}(A)x,\ p_m^{(2)}(A)p_m^{(1)}(A)x,\ \alpha_1 p_m^{(3)}(A)p_m^{(1)}(A)x + \alpha_2 p_m^{(3)}(A)p_m^{(2)}(A)p_m^{(1)}(A)x\},$$

where $\alpha_2 \neq 0$ by the designed filtering (if the previous subspace can be augmented). This list easily extends to $K_j$ for any $j$. Letting

$$\Phi_k(t) = \prod_{l=1}^{k} p_m^{(l)}(t), \tag{5.2}$$

then the last term in $K_j$ contains $\Phi_j(A)x$ with a nonzero coefficient. The filtering is designed such that the subspace contains a significant direction in $\Phi_j(A)x$. Note that $\Phi_j(A)x$ corresponds to an accelerated power method applied to $x$. Because of the Rayleigh–Ritz refinement, there is a Ritz vector from $K_j$ which converges to $q_1$ at least as fast as $\Phi_j(A)x$ does, where the convergence rate for $\Phi_j(A)x$ to $q_1$ is $\frac{\max_{l>1}|\Phi_j(\lambda_l)|}{|\Phi_j(\lambda_1)|}$ under standard conditions [39, 16]. This rate can be considerably faster than the one obtained using a fixed filtering interval. The convergence for the latter eigenvectors follows from deflation; e.g., the second eigenvalue becomes dominant for the matrix $A$ restricted to the subspace orthogonal to $q_1$.

**6. Numerical results and discussion.** We compare the Chebyshev–Davidson method (denoted as ChebyD) with several other Davidson-type and Lanczos-type methods.

The first part of the comparison is done using Matlab. We compare our algorithm with the well-known JD method implemented in the publicly available Matlab code JDQR [14][1] and JDCG [20].[2] The JDCG code is for symmetric eigenproblems; the linear solver used in JDCG is (preconditioned) CG. The JDQR code can solve both symmetric and nonsymmetric problems; GMRES [28] is the default linear solver in JDQR, and it is used for the numerical tests. Since we solve symmetric eigenproblems, it is less costly to use MINRES [21] in the JD method. Moreover, since CG is usually intended for positive definite problems, JDCG does not work as efficiently for indefinite $A$ as for positive definite $A$. So for further comparisons, we implemented the JD method using the Matlab built-in MINRES as the linear solver. This code is denoted

---

[1] Code available at http://www.math.uu.nl/people/sleijpen/JD_software/.

[2] Code available at http://mntek3.ulb.ac.be/pub/docs/jdcg/.

TABLE 6.1

*Silicon quantum dot model $Si_{34}H_{36}$, indefinite. dim = 97569, $k_{want}$ = 100. For ChebyD $m = 20, k_{keep} = 60$; for JDminres #max_le_solve = 20.*

| Method | CPU (sec.) | #iter. | #mvp | $\|\|AV - VD\|\|/\|\|A\|\|_1$ |
|---|---|---|---|---|
| ChebyD | 1204 | 706 | 14806 | 4.16e-11 |
| JDminres | 1968 | 536 | 12306 | 6.44e-11 |
| JDQR | 3734 | 2183(-) | 11850 | 2.73e-13 |
| JDCG | 3597 | 927 | 37899 | 2.90e-12 |
| LOBPCG | 24190 | 5289 | 528900 | 2.98e-10 |

TABLE 6.2

`bcsstk32` *from the NIST Matrix Market, indefinite. dim = 44609, $k_{want}$ = 100. For ChebyD $m = 20, k_{keep} = 60$; for JDminres #max_le_solve = 20.*

| Method | CPU (sec.) | #iter. | #mvp | $\|\|AV - VD\|\|/\|\|A\|\|_1$ |
|---|---|---|---|---|
| ChebyD | 418 | 587 | 12307 | 2.38e-11 |
| JDminres | 850 | 577 | 10360 | 3.55e-11 |
| JDQR | 1186 | 1441(-) | 7639 | 5.80e-14 |
| JDCG | 1250 | 695 | 29810 | 8.86e-13 |
| LOBPCG | 1974 | 686 | 68600 | 9.96e-11 |

TABLE 6.3

*Silicon quantum dot model $Si_{87}H_{76}$, indefinite. dim = 240369. $k_{want}$ = 80. For ChebyD $m = 20, k_{keep} = 80$; for JDminres #max_le_solve = 20.*

| Method | CPU (sec.) | #iter. | #mvp | $\|\|AV - VD\|\|/\|\|A\|\|_1$ |
|---|---|---|---|---|
| ChebyD | 2915 | 493 | 10333 | 3.15e-11 |
| JDminres | 3725 | 497 | 11409 | 2.69e-11 |
| JDQR | 7879 | 2390(-) | 13246 | 2.86e-13 |
| JDCG | 7220 | 720 | 37520 | 2.48e-12 |
| LOBPCG | 13850 | 667 | 53360 | 1.19e-10 |

as JDminres. JDminres is mainly based on Algorithm 4.1, except that step 8a is replaced by a linear equation solve using MINRES. The LOBPCG [15] code[3] is also used for comparison because it is a representative preconditioned eigensolver. We also compared with IRBL [2, 3], but noticed that the IRBL code becomes less competitive than other codes when $k_{want}$ becomes large. Here we report only comparisons with JDQR, JDCG, JDminres, and LOBPCG.

For the test examples listed in Tables 6.1–6.3, we compute the $k_{want}$ smallest eigenvalues and eigenvectors. The maximum subspace dimension is fixed at $2 * k_{want}$ for all methods, except that for LOBPCG it is $3 * k_{want}$. The silicon quantum dot models are available from the University of Florida Sparse Matrix Collection.[4] Figure 6.1 shows the sparsity structure of two test matrices used in Tables 6.1 and 6.2.

The accuracy is reported as $||AV - VD||/||A||_1$, where the diagonal of $D$ contains the $k_{want}$ converged eigenvalues, and $V$ contains the corresponding eigenvectors. The relative convergence tolerance is set to $10^{-10}$ for all methods. For each test problem, the computed eigenvalues are cross validated; i.e., we compute the maximum difference of the eigenvalues computed by different methods. All the differences are found to be less than order $10^{-10}$.

---

[3] Code available at http://www-math.cudenver.edu/~aknyazev/software/CG/toward/lobpcg.m.

[4] http://www.cise.ufl.edu/research/sparse/matrices/.

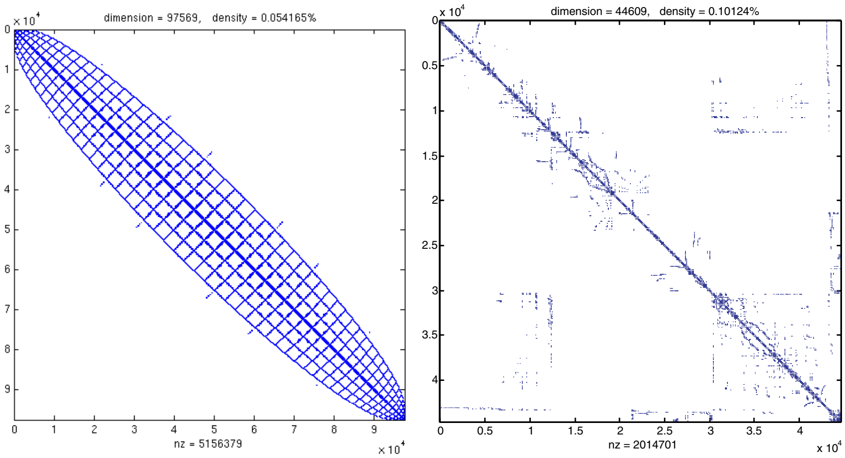


FIG. 6.1. *Structure plots of two test matrices* $Si_{34}H_{36}$ *and* `bcsstk32`.

Initial vector is set as $ones(n, 1)$ for all methods, so that the initial direction is not biased for a certain method. The LOBPCG requires additional $k_{want} - 1$ vectors for the initial block, for which random vectors are used.

In each table, "#iter" counts the total number of the outer loop, "#mvp" is the number of matrix-vector products, and "#max_le_solve" is the maximum inner iteration number for the linear equation solve by MINRES in JDminres. For JDQR, `help jdqr` indicates where #iter and #mvp are stored, but the observed output values from history(:,2) for #iter seem incompatible with the expected values. It is possible that history(:,2) stores both the outer iteration count as well as the inner iteration count, since the resulting number is often much larger than that of JDminres and JDCG. We report #iter for JDQR only for reference, and put a (-) sign to signal the difference. The #mvp for LOBPCG is reported by #iter times the block size (which is $k_{want}$ in LOBPCG).

All the Matlab numerical experiments were performed on an AMD PC with dual Opteron 2.6GHz CPU and 8GB RAM. One of the CPU was dedicated to the computation. The OS used was Red Hat EL4 Linux with kernel version 2.6.9. We used Matlab version 7.2 (R2006a) for the computations.

Note that JDCG and LOBPCG both have preconditioned CG solvers included. However, for the symmetric indefinite problems in Tables 6.1–6.3 (and other test problems not reported here), our experiments showed that both methods work better than their "preconditioned" counterparts with a standard preconditioner such as the incomplete LU. Moreover, for these indefinite problems, it is not clear what preconditioners can be used to accelerate the preconditioned CG solves. Therefore we report comparisons with JDCG and LOBPCG without preconditioned solves. The examples show that in situations where preconditioners are hard to obtain, approaches not relying on solving correction-equations have a clear advantage and can provide effective alternatives.

We recall that the "preconditioning" concept for eigenproblems is quite different from preconditioning for linear equations. The latter tries to reduce the eigenvalue gaps to make the condition number close to 1, while the former tries to introduce more favorable gaps for wanted eigenvalues. This is why in eigenvalue problems, the preconditioned linear solvers are often applied to correction equations, which leads to techniques that exploit shift-and-invert. In essence, a natural "preconditioner" for an eigenvalue problem is a filter that can transform the spectrum in a desired way so as to increase eigenvalue gaps. This "preconditioning" may not need a preconditioned linear solver. In the Chebyshev–Davidson method, we realize the "preconditioning" by dynamically constructing Chebyshev filters to filter the spectrum so that gaps among wanted eigenvalues are properly magnified.

The reported results of ChebyD are typical for the Chebyshev–Davidson method. For all the test runs, it is rather straightforward to select the polynomial degree $m$. The effect of a varying degree $m$ is illustrated in Figure 6.2. As seen from this figure, with $m$ increasing (before it becomes unnecessarily large), the number of iterations decreases, the number of matrix-vector products increases, and the CPU time decreases. Note that the CPU time difference for $m$ from 26 to 47 is not large. The test matrices are the $Si_{10}H_{16}$ and $Si_{34}H_{36}$ silicon quantum dots, but we note that similar behavior is observed for a large number of other test models. In Tables 6.1–6.3 we used $m = 20$ to see how the algorithm performs with a low degree polynomial. A better CPU time for Chebyshev–Davidson can be obtained with a larger $m$. The results in Tables 6.1–6.3 show that even without a fine-tuned $m$, the Chebyshev–Davidson method outperforms other Davidson-type methods.

The numerical results in Tables 6.1–6.3 and Figure 6.2 also show that a smaller #mvp count does not necessarily imply smaller CPU time. Eigenvalue algorithms may require substantial amounts of work not related to matrix-vector products. For example, in Figure 6.2, #mvp increases with $m$ increasing, but because #iter decreases, there is less reorthogonalization cost involved; this explains why the CPU time decreases as $m$ increases. As pointed out in [30], for large sparse eigenproblems where a large number of eigenpairs need to be computed, the total cost can be dominated by the reorthogonalization cost.

Regarding global convergence, Figure 6.3 shows one example where convergence of the Chebyshev–Davidson method is much faster than that of the standard JD approach. However, we would like to mention that for symmetric eigenproblems, a JD method often has good global convergence. For the same example as in Figure 6.3, a fine-tuned value of #max_le_solve for JDminres can make the global convergence of ChebyD and JDminres become similar.

The Chebyshev–Davidson algorithm was also implemented in Fortran95; its parallel version has been integrated into an electronic structure calculation package called PARSEC (pseudopotetial algorithm for real-space electronic calculations). PARSEC uses real-space pseudopotential implementation of density functional theory methods. The original ideas behind PARSEC date back to the early 1990s [8, 9]. Originally, PARSEC had three diagonalization methods: a preconditioned Davidson method [29, 35] called Diagla, the symmetric eigensolver from ARPACK [33, 17], and the thick-restart Lanczos method (TRLan) [40, 42]. The Chebyshev–Davidson algorithm was subsequently integrated into PARSEC (around October, 2005). Due to its efficiency and robustness, it was quickly adopted as the default eigensolver by our collaborators in material science. In the latest version of PARSEC, a true diagonalization is performed only at the first step of the self-consistent loop, after which diagonalizations are replaced by a nonlinear Chebyshev-filtered-subspace (CheFS) method [47, 46]. Nevertheless, the first diagonalization step can still be highly challenging. This is because a relatively complex material system can contain several thousand atoms, in which case the dimension of the discretized Hamiltonians can easily exceed several millions. Even more challenging is the fact that the number of eigenpairs needed

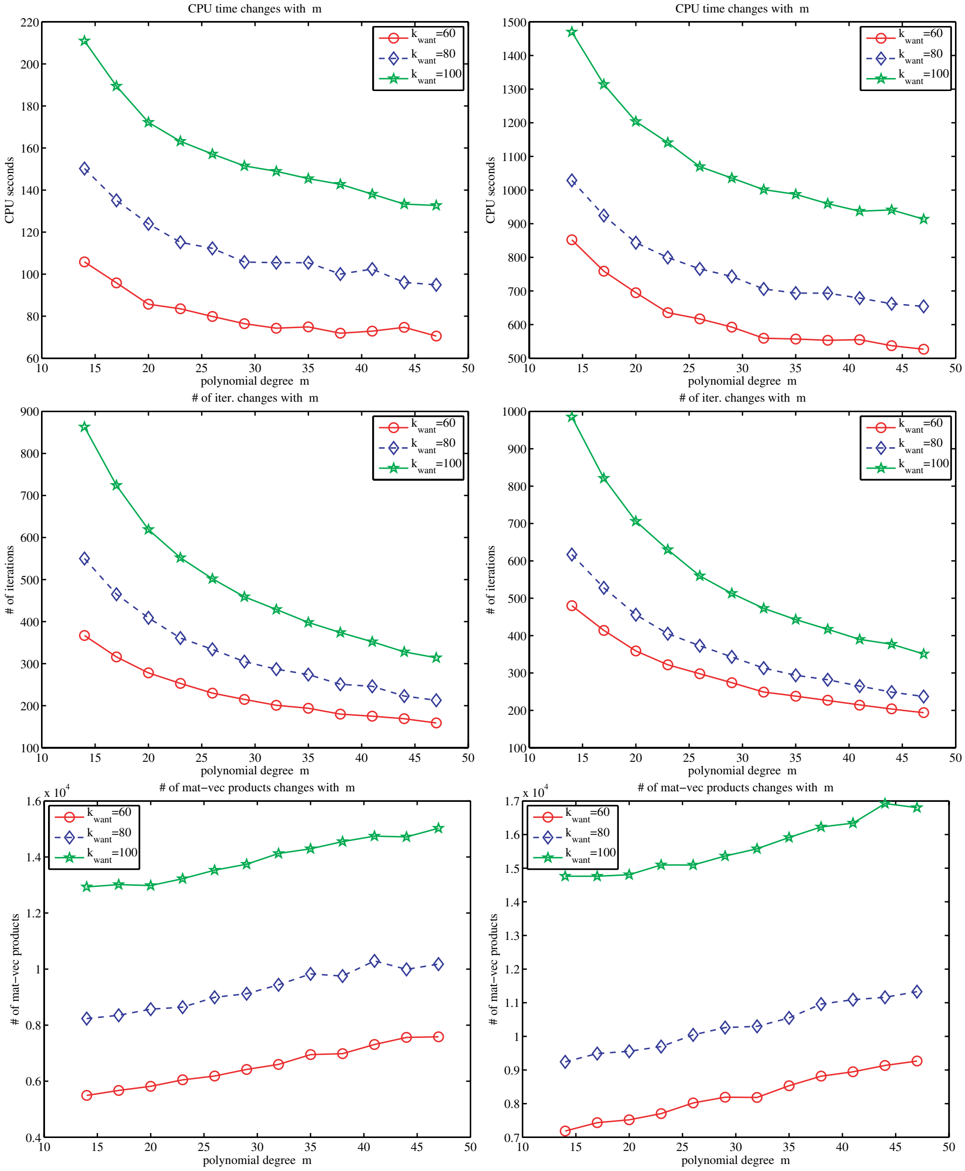


FIG. 6.2. *Changes in CPU time, number of iterations, and number of matrix-vector products with a varying polynomial degree m. Figures on the left are for quantum dot* $Si_{10}H_{16}$*, with* $n = 17077$*. Figures on the right are for quantum dot* $Si_{34}H_{36}$*, with* $n = 97569$*. The* $m$ *is varied as* $m = 14 : 3 : 47$ *in Matlab notation. For each model, the same initial vector* $ones(n, 1)$ *is used for each* $m$*. The number of vectors to keep during restart is simply set as* $k_{keep} = 60$ *for all these tests. Three cases where* $k_{want} = 60, 80, 100$ *are demonstrated.*

is proportional to the number of valence electrons in the atoms, which commonly exceed several thousand. In these cases, high memory demand is clearly a concern. Moreover, eigenvalue algorithms that are efficient for exterior eigenvalues can have problems converging for interior eigenvalues.

Table 6.4 shows the dimension of the discretized Hamiltonians and the number of needed eigenpairs for four silicon nanocrystals and two metallic (iron) clusters. The

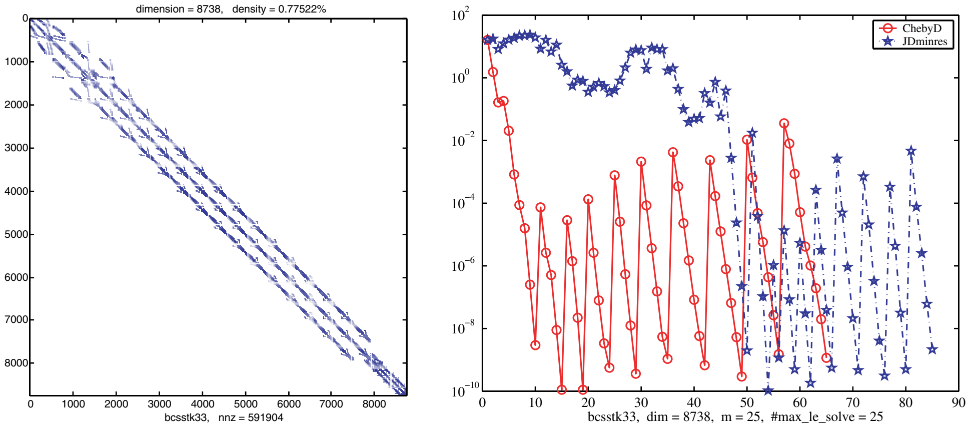


FIG. 6.3. *The matrix* `bcsstk33` *is from the NIST Matrix Market. Structure plot is on the left. On the right is the residual norm plot for the first* 10 *smallest eigenvalues. This shows that ChebyD can have much better global convergence than JDminres. Here* $m = 25$ *for ChebyD and* $\#max\_le\_solve = 25$ *for JDminres. The initial vector used is* $ones(n, 1)$ *for both methods.*

TABLE 6.4

*The Chebyshev–Davidson method applied to compute* $k_{want}$ *number of eigenvalues and eigenvectors. For the silicon nanocrystals, the polynomial degree is* $m = 17$*; for the iron clusters,* $m = 20$*. The computations are performed on the SGI Altix cluster (*1.6*GHz per processor) at the Minnesota Supercomputing Institute.*

| Material | Matrix dimension $n$ | $k_{want}$ | # Processors | CPU hours |
|---|---|---|---|---|
| $Si_{2713}H_{828}$ | 1074080 | 5843 | 16 | 7.83 |
| $Si_{4001}H_{1012}$ | 1472440 | 8511 | 16 | 18.63 |
| $Si_{6047}H_{1308}$ | 2144432 | 12751 | 32 | 45.11 |
| $Si_{9041}H_{1860}$ | 2992832 | 19015 | 48 | 102.12 |
| $Fe_{326}$ | 2985992 | 3912 | 24 | 11.62 |
| $Fe_{360}$ | 3262312 | 4320 | 24 | 16.55 |

reported CPU time is what the Chebyshev–Davidson method used to finish the first step diagonalization in the self-consistent loop.

Physical significance of the numerical results are discussed in [37, 38]. In [37] we report the largest iron-cluster first principle DFT simulations that have been published. The results are used to clarify a decade-old controversy regarding the dependence of magnetic moment on the size of iron clusters. As to first principles DFT calculations on silicon nanocrystals, previously reported results seem not have gone beyond 2000 atoms; in contrast, we were able to do first principle calculations on a sequence of silicon nanocrystals with up to 10,000 atoms [46, 7].

Although success in these challenging DFT calculations depends more on the nonlinear CheFS method, we must mention that the Chebyshev–Davidson method plays a crucial role in the computations since it provides the CheFS method with a desired initial subspace. A suitable initial subspace can substantially reduce the number of iterations required for the CheFS method to reach self-consistency (convergence).

The other three eigensolvers (Diagla, ARPACK, and TRLan) in PARSEC were also used for computing initial subspaces, but we noticed that they became impractical for the largest material systems in Table 6.4, in terms of both memory requirement and convergence speed. In comparison, Diagla is quite efficient when $n$ and $k_{want}$ are moderate, but it becomes slower than ARPACK and TRLan when $n$ and $k_{want}$ become large. TRLan is the fastest among these three solvers; it is observed in [47] to be about twice as fast as ARPACK because of the reduced reorthogonalization. But both TRLan and ARPACK require too much memory because of the requirement that the maximum subspace dimension be around $2k_{want}$.

To address the huge memory demand related to standard restart when $n$ and $k_{want}$ are large, we introduced an [illegible] technique into the Chebyshev–Davidson algorithm. The [illegible] is the same as the standard restart, but the [illegible] corresponds to a standard restart restricted to an inner subspace with dimension much smaller than $k_{want}$. This reduces the maximum dimension of the outer subspace from $2k_{want}$ to $k_{want}$. Therefore the Chebyshev–Davidson algorithm requires about half the memory required by a method with only standard restart. It did not have a memory requirement problem for all the materials reported in Table 6.4. More details about the [illegible] may be found in [44].

As to CPU time, we compared the Chebyshev–Davidson method with TRLan on the two smallest nanocrystals in Table 6.4. Using the same number of CPU nodes, TRLan spent 8.65 CPU hours on $Si_{2713}H_{828}$ and 34.99 CPU hours on $Si_{4001}H_{1012}$ for the first step diagonalization. The comparison is not completely fair since we employed an additional trick in the Chebyshev–Davidson routine, which corresponds to a subspace filtering step so that the last few basis vectors are only approximate eigenvectors. The number of these vectors not converged to full accuracy is bounded above by the dimension of the inner subspace used for inner restart. It is rather straightforward to add this subspace filtering step inside a Davidson-type iteration. Both this trick and the [illegible] are due to the remarkable flexibility of a Davidson-type method in adjusting basis vectors. A Lanczos-type method does not have this flexibility because of the need to keep a Krylov structure. In TRLan all the basis vectors are converged to the same full accuracy, which can be too costly since high accuracy is often not necessary for the last few vectors in the subspace, especially when the diagonalization is performed at the first step of the self-consistent loop to provide an initial subspace.

We also mention that the adaptive Chebyshev filter (based on [26, 27]) and the choice of bounds to achieve efficient filtering, as presented in this paper, are essential to the development of the nonlinear CheFSI method in [47, 46].

## 7. Conclusion.

A Chebyshev–Davidson algorithm has been presented for solving large symmetric eigenvalue problems. It essentially consists of filtering out the unwanted portion of the spectrum by using adaptive Chebyshev polynomials of the matrix. Comparisons with existing Davidson- and Lanczos-type methods show that the Chebyshev–Davidson method is efficient and robust.

Advantages of the Chebyshev filtering approach include not requiring correction-equations (hence no preconditioned linear solves are necessary), and robust global convergence because of the intervalwise filtering. The Chebyshev filters are easily controllable within the Davidson-type framework, and thus they can be conveniently tuned to filter the full spectrum in the desired way to accelerate global convergence.

**Acknowledgments.** We thank Professor Calvetti for her constructive comments, especially the suggestion on testing the effect of the polynomial degree in the Chebyshev–Davidson algorithm, which improved our understanding of this parameter.

## REFERENCES

[1] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminara, *A comparison of eigensolvers for large-scale* 3*d model analysis using AMG-preconditioned iterative methods*, Int. J. Numer. Methods Engrg., 64 (2005), pp. 204–236.

[2] J. Baglama, D. Calvetti, and L. Reichel, *IRBL: An implicitly restarted block-Lanczos method for large-scale Hermitian eigenproblems*, SIAM J. Sci. Comput., 24 (2003), pp. 1650–1677.

[3] J. Baglama, D. Calvetti, and L. Reichel, *irbleigs: A MATLAB program for computing a few eigenpairs of a large sparse Hermitian matrix*, ACM Trans. Math. Softw., 5 (2003), pp. 337–348.

[4] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Software Environ. Tools 11, SIAM, Philadelphia, PA, 2000.

[5] J. H. Brandts, *Solving eigenproblems: From Arnoldi via Jacobi-Davidson to the Riccati method*, in Numerical Methods and Applications, Lecture Notes in Comput. Sci. 2542, Comput. Sci., Springer, New York, 2003, pp. 167–173.

[6] J. Chelikowsky and Y. Saad, *Electronic structure of clusters and nanocrystals*, in Handbook of Theoretical and Computational Nanotechnology, M. Rieth and W. Schommers, eds., American Scientific Publishers, Stevenson Ranch, CA, to appear.

[7] J. R. Chelikowsky, M. L. Tiago, Y. Saad, and Y. Zhou, *Algorithms for the evolution of electronic properties in nanocrystals*, Comp. Phys. Comm., 177 (2007), pp. 1–5.

[8] J. R. Chelikowsky, N. Troullier, and Y. Saad, *Finite-difference-pseudopotential method: Electronic structure calculations without a basis*, Phys. Rev. Lett., 72 (1994), pp. 1240–1243.

[9] J. R. Chelikowsky, N. Troullier, K. Wu, and Y. Saad, *Higher-order finite-difference pseudopotential method: An application to diatomic molecules*, Phys. Rev. B, 50 (1994), pp. 11355–11364.

[10] J. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 772–795.

[11] E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, J. Comput. Phys., 17 (1975), pp. 87–94.

[12] R. M. Dreizler and E. K. U. Gross, *Density Functional Theory: An Approach to the Quantum Many-Body Problem*, Springer-Verlag, Berlin, 1990.

[13] Y. T. Feng, *An integrated multigrid and Davidson method for very large scale symmetric eigenvalue problems*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 3543–3563.

[14] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils*, SIAM J. Sci. Comput., 20 (1998), pp. 94–125.

[15] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

[16] R. B. Lehoucq, *Implicitly restarted Arnoldi methods and subspace iteration*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 551–562.

[17] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software Environ. Tools 6, SIAM, Philadelphia, 1998; software available online at: http://www.caam.rice.edu/software/ARPACK/.

[18] R. M. Martin, *Electronic Structure : Basic Theory and Practical Methods*, Cambridge University Press, Cambridge, UK, 2004.

[19] R. B. Morgan and D. S. Scott, *Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 817–825.

[20] Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem*, Numer. Linear Algebra Appl., 9 (2002), pp. 21–44.

[21] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[22] B. N. Parlett, *The Rayleigh quotient iteration and some generalizations for nonnormal matrices*, Math. Comp., 28 (1974), pp. 679–693.

[23] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Appl. Math. 20, SIAM, Philadelphia, PA, 1997.

[24] H. Rutishauser, *Computational aspects of F. L. Bauer's simultaneous iteration method*, Numer. Math., 13 (1969), pp. 4–13.

[25] H. Rutishauser, *Simultaneous iteration method for symmetric matrices*, in Handbook for Automatic Computation (Linear Algebra), J. H. Wilkinson and C. Reinsh, eds., Springer-Verlag, 1971, vol. II, pp. 284–302.

[26] Y. Saad, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.

[27] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, John Wiley, New York, 1992.

[28] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[29] Y. Saad, A. Stathopoulos, J. Chelikowsky, K. Wu, and S. Öğüt, *Solution of large eigenvalue problems in electronic structure calculations*, BIT, 36 (1996), pp. 563–578.

[30] Y. Saad, Y. Zhou, C. Bekas, M. Tiago, and J. Chelikowsky, *Diagonalization methods in PARSEC*, Phys. Status Solidi (B), 243 (2006), pp. 2188–2197.

[31] G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems*, BIT, 36 (1996), pp. 595–633.

[32] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi–Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425.

[33] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.

[34] D. C. Sorensen and C. Yang, *Accelerating the Lanczos Algorithm via Polynomial Spectral Transformations*, Technical Report TR97-29, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1997.

[35] A. Stathopoulos, S. Öğüt, Y. Saad, J. Chelikowsky, and H. Kim, *Parallel methods and tools for predicting materials properties*, Comput. Sci. Eng., 2 (2000), pp. 19–32.

[36] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.

[37] M. L. Tiago, Y. Zhou, M. Alemany, Y. Saad, and J. R. Chelikowsky, *Evolution of magnetism in iron from the atom to the bulk*, Phys. Rev. Lett., 97 (2006), paper 147201.

[38] M. L. Tiago, Y. Zhou, Y. Saad, and J. R. Chelikowsky, *Electronic Properties and Energetics of Nanometer-size Silicon Nanocrystals*, Technical report, ICES, University of Texas/Austin, Austin, TX, in preparation.

[39] D. S. Watkins and L. Elsner, *Convergence of algorithms of decomposition type for the eigenvalue problem*, Linear Algebra Appl., 41 (1991), pp. 19–47.

[40] K. Wu, A. Canning, H. D. Simon, and L.-W. Wang, *Thick-restart Lanczos method for electronic structure calculations*, J. Comput. Phys., 154 (1999), pp. 156–173.

[41] K. Wu, Y. Saad, and A. Stathopoulos, *Inexact Newton preconditioning techniques for large symmetric eigenvalue problems*, Electron. Trans. Numer. Anal., 7 (1998), pp. 202–214.

[42] K. Wu and H. Simon, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.

[43] Y. Zhou, *Studies on Jacobi-Davidson, Rayleigh quotient iteration, inverse iteration generalized Davidson and Newton updates*, Numer. Linear Algebra Appl., 13 (2006), pp. 621–642.

[44] Y. Zhou, *A Block Chebyshev–Davidson Method with Inner-Outer Restart for Large Eigenvalue Problems*, Technical report, Department of Mathematics, Southern Methodist University, Dallas, TX, in preparation.

[45] Y. Zhou and Y. Saad, *Block Krylov-Schur Method for Large Symmetric Eigenvalue Problems*, Technical report 2004/215, Minnesota Supercomputing Institute, University of Minnesota, 2004.

[46] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Parallel self-consistent-field calculations using Chebyshev-filtered subspace acceleration*, Phys. Rev. E, 74 (2006), paper 066704.

[47] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Self-consistent-field calculation using Chebyshev-filtered subspace iteration*, J. Comput. Phys., 219 (2006), pp. 172–184.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 972–981

# JOINT SINGULAR VALUE DISTRIBUTION OF TWO CORRELATED RECTANGULAR GAUSSIAN MATRICES AND ITS APPLICATION[*]

SHUANGQUAN WANG[†] AND ALI ABDI[†]

**Abstract.** Let $\mathbf{H} = (h_{ij})$ and $\mathbf{G} = (g_{ij})$ be two $m \times n$, $m \leq n$, *rectangular* random matrices, each with independently and identically distributed complex zero-mean unit-variance Gaussian entries, with correlation between any two elements given by $\mathbb{E}[h_{ij} g_{pq}^{\star}] = \rho\, \delta_{ip} \delta_{jq}$ such that $|\rho| < 1$, where $^{\star}$ denotes the complex conjugate and $\delta_{ij}$ is the Kronecker delta. Assume $\{s_k\}_{k=1}^{m}$ and $\{r_l\}_{l=1}^{m}$ are unordered singular values of $\mathbf{H}$ and $\mathbf{G}$, respectively, and $s$ and $r$ are randomly selected from $\{s_k\}_{k=1}^{m}$ and $\{r_l\}_{l=1}^{m}$, respectively. In this paper, exact analytical closed-form expressions are derived for the joint probability distribution function (PDF) of $\{s_k\}_{k=1}^{m}$ and $\{r_l\}_{l=1}^{m}$ using an Itzykson–Zuber-type integral as well as the joint marginal PDF of $s$ and $r$ by a biorthogonal polynomial technique. These PDFs are of interest in multiple-input multiple-output wireless communication channels and systems.

**Key words.** correlated complex random matrices, joint singular value distribution, biorthogonal polynomials

**AMS subject classifications.** 15A52, 15A18, 62E15, 33C45

**DOI.** 10.1137/060652907

## 1. Introduction.

Random singular values have found numerous applications such as hypothesis testing and principal component analysis in statistics [14], nuclear energy levels and level spacing in nuclear physics [12], quantum chromodynamics [20], and calculation of the multiple-input multiple-output (MIMO) channel capacity in wireless communications [18]. The singular value distribution of a [illegible] Gaussian random matrix is given in [16]. For a single chiral random matrix, where a square Gaussian random matrix and its Hermitian sit on the off diagonal, the result is reported in [6]. However, the joint singular value distribution of [illegible] Gaussian random matrices has received less attention so far, although it has important applications in wireless MIMO communications, say, the second-order statistics of the [illegible] [22, Chap. 4] [25] and instantaneous mutual information [22, Chaps. 3 and 4] [23, 24, 26].

To the best of our knowledge, correlated square random matrices have been studied to some extent [4,12,13], where only Hermitian matrices were considered. Different from [4,12,13], we consider the situation where the elements, with the same indices, of the two rectangular complex Gaussian random matrices are correlated by a [illegible] number and derive exact analytical closed-form expressions for the joint probability distribution function (PDF) of their singular values.

This paper is organized as follows. Section 2 introduces the two rectangular complex Gaussian random matrices. The joint PDFs of singular values are studied in section 3 using an Itzykson–Zuber-type integral. The joint marginal PDF of singular values is derived in section 4, and its application to wireless MIMO communications is presented in section 5. Finally, concluding remarks are summarized in section 6.

---

[*] Received by the editors February 24, 2006; accepted for publication (in revised form) by H. J. Werner May 25, 2007; published electronically October 5, 2007.
http://www.siam.org/journals/simax/29-3/65290.html

[†] Center for Wireless Communications and Signal Processing Research (CWCSPR), Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 (shuangquan.wang@njit.edu, shuangquan@nec-labs.com, ali.abdi@njit.edu). The current address for the first author is NEC Laboratories America, Inc., Princeton, NJ 08540 .

[illegible] $\cdot^\dagger$ is reserved for matrix Hermitian, $\cdot^T$ for matrix transpose, $\cdot^\star$ for complex conjugate, $\mathrm{tr}[\cdot]$ for the trace of a matrix, $\jmath$ for $\sqrt{-1}$, $\mathbb{E}[\cdot]$ for mathematical expectation, $\mathbf{I}_m$ for the $m \times m$ identity matrix, $\otimes$ for the Kronecker product, and $\Re[\cdot]$ and $\Im[\cdot]$ for the real and imaginary parts of a complex number, respectively. In addition, $\mathrm{diag}(\mathbf{s})$ denotes a diagonal matrix with $\mathbf{s}$ on the main diagonal, $t \in [m, n]$ implies that $t$, $m$, and $n$ are all integers such that $m \leq t \leq n$, with $m \leq n$, and $\det|x_{kl}|$ is the determinant of the matrix, where $x_{kl}$ resides on the $k$th row and $l$th column. Moreover, lowercase bold letters represent row vectors, whereas uppercase bold letters are used for matrices. Finally $\mathcal{CN}$ means complex normal, and $\mathrm{vec}(\cdot)$ stacks all of the columns of its matrix argument into one tall column vector.

## 2. Problem description.

There are two $m \times n$ random matrices $\mathbf{H} = (h_{ij})$ and $\mathbf{G} = (g_{ij})$, $i \in [1, m]$, $j \in [1, n]$, each with independently and identically distributed (i.i.d.) complex zero-mean unit-variance Gaussian entries, i.e., $\mathbb{E}[h_{ij}] = \mathbb{E}[g_{ij}] = 0\ \forall i, j$, $\mathbb{E}[h_{ij}h_{pq}^\star] = \mathbb{E}[g_{ij}g_{pq}^\star] = \delta_{ip}\delta_{jq}$, where the Kronecker symbol $\delta_{ij}$ is 1 or 0 when $i = j$ or $i \neq j$. Therefore $\mathbf{H}, \mathbf{G} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{mn})$. Moreover, the correlation between the two random matrices is given by

$$\mathbb{E}[h_{ij}g_{pq}^\star] = \rho\,\delta_{ip}\delta_{jq} \quad \forall i, j, p, q, \tag{2.1}$$

where $\rho = |\rho|e^{\jmath\theta}$ is a complex number, with $|\rho| < 1$.

Without loss of generality, we assume $m \leq n$ and set $\nu = n - m$. Based on the singular value decomposition (SVD), $\mathbf{H}$ and $\mathbf{G}$ can be, respectively, diagonalized as [8]

$$\mathbf{H} = \mathbf{U}\mathbf{S}\mathbf{V}^\dagger, \tag{2.2}$$

$$\mathbf{G} = \widetilde{\mathbf{U}}\mathbf{R}\widetilde{\mathbf{V}}^\dagger, \tag{2.3}$$

where $\mathbf{S} = \big[\mathrm{diag}(\mathbf{s})\ \mathbf{0}\big]$ and $\mathbf{R} = \big[\mathrm{diag}(\mathbf{r})\ \mathbf{0}\big]$, with $\mathbf{s} = [s_1, s_2, \ldots, s_m]$ and $\mathbf{r} = [r_1, r_2, \ldots, r_m]$, respectively.

We assume that the singular values of $\mathbf{G}$, $r_1, r_2, \ldots, r_m$, are unordered and the singular values of $\mathbf{H}$, $s_1, s_2, \ldots, s_m$, are also unordered. Now we would like to know the joint PDF of $\{r_l\}_{l=1}^m$ and $\{s_l\}_{l=1}^m$. Moreover, with $r$ randomly selected from $r_1, r_2, \ldots, r_m$, and $s$ randomly selected from $s_1, s_2, \ldots, s_m$, it is of interest to derive the joint PDF of $r$ and $s$ as well. These two PDFs are derived in sections 3 and 4, respectively.

## 3. Joint PDF of $\{s_l\}_{l=1}^m$ and $\{r_l\}_{l=1}^m$.

LEMMA 3.1 (joint PDF of $\mathbf{H}$ and $\mathbf{G}$). [illegible] $\mathbf{H}, \mathbf{G} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{mn})$ [illegible] $\mathbf{H}$ [illegible] $\mathbf{G}$ [illegible] (2.1) [illegible] $\mathbf{H}$ [illegible] $\mathbf{G}$ [illegible]

$$p(\mathbf{H}, \mathbf{G}) = \frac{1}{\pi^{2mn}\left(1 - |\rho|^2\right)^{mn}} \exp\left[-\frac{\mathrm{tr}\big(\mathbf{H}\mathbf{H}^\dagger + \mathbf{G}\mathbf{G}^\dagger - \rho^\star\mathbf{H}\mathbf{G}^\dagger - \rho\mathbf{G}\mathbf{H}^\dagger\big)}{1 - |\rho|^2}\right]. \tag{3.1}$$

[illegible] We set $\mathbf{h} = \mathrm{vec}(\mathbf{H})$, $\mathbf{g} = \mathrm{vec}(\mathbf{G})$, and $\mathbf{x} = [\mathbf{h}^T\ \mathbf{g}^T]^T$. Based on $\mathbf{H}, \mathbf{G} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{mn})$ and (2.1), we have the mean and covariance matrix of $\mathbf{x}$ as $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ and $\Sigma_\mathbf{x} = \Sigma_\tau \otimes \mathbf{I}_{mn}$, with $\Sigma_\tau = \left[\begin{smallmatrix} 1 & \rho \\ \rho^\star & 1 \end{smallmatrix}\right]$, respectively. Therefore the PDF of $\mathbf{x}$ is given by [11]

$$p(\mathbf{x}) = \frac{1}{\pi^{2mn}\det|\Sigma_\mathbf{x}|} \exp\left(-\mathbf{x}^\dagger\Sigma_\mathbf{x}^{-1}\mathbf{x}\right), \tag{3.2}$$

where $\det|\Sigma_\mathbf{x}| = (\det|\Sigma_\tau|)^{mn} = \left(1 - |\rho|^2\right)^{mn}$.

With $\Sigma_\tau^{-1} = \frac{1}{1-|\rho|^2}\begin{bmatrix} 1 & -\rho \\ -\rho^\star & 1 \end{bmatrix}$, we obtain $\Sigma_\mathbf{x}^{-1} = \Sigma_\tau^{-1} \otimes \mathbf{I}_{mn} = \frac{1}{1-|\rho|^2}\begin{bmatrix} \mathbf{I}_{mn} & -\rho\mathbf{I}_{mn} \\ -\rho^\star\mathbf{I}_{mn} & \mathbf{I}_{mn} \end{bmatrix}$. Therefore $\mathbf{x}^\dagger\Sigma_\mathbf{x}^{-1}\mathbf{x}$ in (3.2) can be rewritten as

(3.3)
$$\begin{aligned}\mathbf{x}^\dagger\Sigma_\mathbf{x}^{-1}\mathbf{x} &= \operatorname{tr}\left(\Sigma_\mathbf{x}^{-1}\mathbf{x}\mathbf{x}^\dagger\right) = \operatorname{tr}\left(\frac{1}{1-|\rho|^2}\begin{bmatrix} \mathbf{I}_{mn} & -\rho\mathbf{I}_{mn} \\ -\rho^\star\mathbf{I}_{mn} & \mathbf{I}_{mn} \end{bmatrix}\begin{bmatrix} \mathbf{h}\mathbf{h}^\dagger & \mathbf{h}\mathbf{g}^\dagger \\ \mathbf{g}\mathbf{h}^\dagger & \mathbf{g}\mathbf{g}^\dagger \end{bmatrix}\right), \\ &= \frac{\operatorname{tr}\left(\mathbf{h}\mathbf{h}^\dagger + \mathbf{g}\mathbf{g}^\dagger - \rho^\star\mathbf{h}\mathbf{g}^\dagger - \rho\mathbf{g}\mathbf{h}^\dagger\right)}{1-|\rho|^2} = \frac{\operatorname{tr}\left(\mathbf{H}\mathbf{H}^\dagger + \mathbf{G}\mathbf{G}^\dagger - \rho^\star\mathbf{H}\mathbf{G}^\dagger - \rho\mathbf{G}\mathbf{H}^\dagger\right)}{1-|\rho|^2},\end{aligned}$$

where $\operatorname{tr}\left(\mathbf{A}\mathbf{B}^\dagger\right) = \operatorname{vec}(\mathbf{B})^\dagger\operatorname{vec}(\mathbf{A}) = \operatorname{tr}\left[\operatorname{vec}(\mathbf{A})\operatorname{vec}(\mathbf{B})^\dagger\right]$ [7] is used in the last "=" of (3.3). Substitution of (3.3) into (3.2) leads to (3.1). □

From (2.2), we know that the unitary matrix pair $(\mathbf{U}, \mathbf{V})$ parameterizes the coset space $\mathcal{U}(m) \times \mathcal{U}(n) / [\mathcal{U}(1)]^m$, where $\mathcal{U}(p)$ is the unitary group of order $p$, and the integration measure $d[\mathbf{H}] = \prod_{i=1}^m\prod_{j=1}^n d[\Re h_{ij}]\,d[\Im h_{ij}]$ can be represented by [9]

$$d[\mathbf{H}] = \Omega J(\mathbf{s})d[\mathbf{s}]d\mu(\mathbf{U},\mathbf{V}), \tag{3.4}$$

where $J(\mathbf{s}) = \triangle^2(\mathbf{s}^2)\prod_{k=1}^m s_k^{2\nu+1}$ with the $m$-dimensional Vandermonde determinant $\triangle(\mathbf{s}^2) = \det|s_k^{2(l-1)}| = \prod_{k>l}(s_k^2 - s_l^2)$ and $\triangle^2(\cdot) = [\triangle(\cdot)]^2$, $d[\mathbf{s}] = \prod_{l=1}^m ds_l$, $d\mu(\mathbf{U},\mathbf{V})$ is the Haar measure of $\mathcal{U}(m) \times \mathcal{U}(n) / [\mathcal{U}(1)]^m$ [9], and the constant $\Omega$ is given by [9,15]

$$\Omega = \frac{2^m\pi^{mn}}{\prod_{j=1}^m j!(j+\nu-1)!} = \frac{2^m\pi^{mn}}{m!\prod_{j=0}^{m-1} j!(j+\nu)!}. \tag{3.5}$$

Similarly, we have

$$d[\mathbf{G}] = \Omega J(\mathbf{r})d[\mathbf{r}]d\mu(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}}), \tag{3.6}$$

where $J(\mathbf{r}) = \triangle^2(\mathbf{r}^2)\prod_{k=1}^m r_k^{2\nu+1}$ with the $m$-dimensional Vandermonde determinant $\triangle(\mathbf{r}^2) = \det|r_k^{2(l-1)}| = \prod_{k>l}(r_k^2 - r_l^2)$ and $d[\mathbf{r}] = \prod_{l=1}^m dr_l$.

In order to obtain the joint PDF of $\{r_l\}_{l=1}^m$ and $\{s_l\}_{l=1}^m$, we need the following proposition.

Proposition 3.2 (an Itzykson–Zuber-type integral [9, equation (31)]).

$$\begin{aligned}&\int d\mu(\mathbf{U},\mathbf{V})\exp\left\{-\frac{\operatorname{tr}\left[(\mathbf{H}-\mathbf{G})(\mathbf{H}-\mathbf{G})^\dagger\right]}{t}\right\} \\ &= \frac{2^m\pi^{mn}t^{mn-m}\det\left|\exp\left(-\frac{s_k^2+r_l^2}{t}\right)I_\nu\left(\frac{2s_kr_l}{t}\right)\right|}{m!\Omega\triangle(\mathbf{s}^2)\triangle(\mathbf{r}^2)\prod_{k=1}^m(s_kr_k)^\nu},\end{aligned} \tag{3.7}$$

[illegible] $\Omega$ [illegible] (3.5) [illegible] $I_k(z) = \frac{1}{\pi}\int_0^\pi e^{z\cos\theta}\cos(k\theta)$ [illegible] $\theta$ [illegible] $k$ [illegible]

Theorem 3.3. [illegible] $\mathbf{H}$ [illegible] $\mathbf{G}$ [illegible]

$$p(\mathbf{s},\mathbf{r}) = \frac{\exp\left(-\frac{\sum_{k=1}^m s_k^2+r_k^2}{1-|\rho|^2}\right)\triangle(\mathbf{s}^2)\triangle(\mathbf{r}^2)\prod_{k=1}^m(s_kr_k)^{\nu+1}\det\left|I_\nu\left(\frac{2|\rho|s_kr_l}{1-|\rho|^2}\right)\right|}{2^{-2m}m!m!\prod_{j=0}^{m-1}j!(j+\nu)!|\rho|^{mn-m}(1-|\rho|^2)^m}. \tag{3.8}$$

[illegible] By combining (3.1) with (3.4) and (3.6), we obtain

$$p(\mathbf{s},\mathbf{r}) = \frac{\Omega^2 J(\mathbf{s})J(\mathbf{r})}{\pi^{2mn}(1-|\rho|^2)^{mn}}\Phi(\mathbf{s},\mathbf{r}), \tag{3.9}$$

where

$$
\begin{aligned}
\Phi(\mathbf{s},\mathbf{r}) &= \int d\mu(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}}) \int d\mu(\mathbf{U},\mathbf{V}) \exp\left[-\frac{\operatorname{tr}\left(\mathbf{H}\mathbf{H}^\dagger + \mathbf{G}\mathbf{G}^\dagger - \rho^\star \mathbf{H}\mathbf{G}^\dagger - \rho \mathbf{G}\mathbf{H}^\dagger\right)}{1-|\rho|^2}\right] \\
&= \int d\mu(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}}) \int d\mu(\mathbf{U},\mathbf{V}) \exp\left\{-\frac{\operatorname{tr}\left[(\mathbf{H}-\rho\mathbf{G})(\mathbf{H}-\rho\mathbf{G})^\dagger\right]}{1-|\rho|^2} - \operatorname{tr}(\mathbf{G}\mathbf{G}^\dagger)\right\} \\
&= \int d\mu(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}})\, e^{-\operatorname{tr}(\mathbf{G}\mathbf{G}^\dagger)} \int d\mu(\mathbf{U},\mathbf{V}) \exp\left\{-\frac{\operatorname{tr}\left[(\mathbf{H}-\rho\mathbf{G})(\mathbf{H}-\rho\mathbf{G})^\dagger\right]}{1-|\rho|^2}\right\} \\
&= \int d\mu(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}}) \frac{e^{-\sum_{k=1}^m r_k^2}(1-|\rho|^2)^{mn-m} \det\left| e^{-\frac{s_k^2+|\rho|^2 r_l^2}{1-|\rho|^2}} I_\nu\!\left(\frac{2|\rho| s_k r_l}{1-|\rho|^2}\right)\right|}{2^{-m} m! \pi^{-mn} \Omega \triangle(\mathbf{s}^2) \triangle(|\rho|^2 \mathbf{r}^2) \prod_{k=1}^m (|\rho| s_k r_k)^\nu} \\
&= \frac{(1-|\rho|^2)^{mn-m} \exp\left(-\frac{\sum_{k=1}^m s_k^2 + r_k^2}{1-|\rho|^2}\right) \det\left| I_\nu\!\left(\frac{2|\rho| s_k r_l}{1-|\rho|^2}\right)\right|}{2^{-m} m! \pi^{-mn} \Omega |\rho|^{mn-m} \triangle(\mathbf{s}^2) \triangle(\mathbf{r}^2) \prod_{k=1}^m (s_k r_k)^\nu}.
\end{aligned}
\tag{3.10}
$$

Derivation of the second and third lines of (3.10) are straightforward. The fourth line comes from

$$\rho\mathbf{G} = \widehat{\mathbf{U}}\widehat{\mathbf{R}}\widehat{\mathbf{V}}^\dagger, \tag{3.11}$$

with $\widehat{\mathbf{R}} = |\rho|\mathbf{R}$, and Proposition 3.2 with the replacements $t \to 1-|\rho|^2$ and $\mathbf{G} \to \rho\mathbf{G}$. The last line is based on the convention that $\int d\mu(\widetilde{\mathbf{U}},\widetilde{\mathbf{V}}) = 1$ [9]. Plugging (3.5) and the last line of (3.10) into (3.9), we obtain (3.8). $\square$

By relating the eigenvalues of $\mathbf{G}\mathbf{G}^\dagger$ to the singular values of $\mathbf{G}$ through $\alpha_l = r_l^2$, $l \in [1,m]$, and the eigenvalues of $\mathbf{H}\mathbf{H}^\dagger$ to the singular values of $\mathbf{H}$ through $\beta_l = s_l^2$, $l \in [1,m]$, we can derive the joint PDF of $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_m]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_m]$, presented in the following corollary.

COROLLARY 3.4. [illegible] $\mathbf{H}\mathbf{H}^\dagger$ [illegible] $\mathbf{G}\mathbf{G}^\dagger$ [illegible]

$$p(\boldsymbol{\beta},\boldsymbol{\alpha}) = \frac{\exp\left(-\frac{\sum_{k=1}^m \beta_k + \alpha_k}{1-|\rho|^2}\right) \triangle(\boldsymbol{\beta}) \triangle(\boldsymbol{\alpha}) \prod_{k=1}^m (\sqrt{\beta_k \alpha_k})^\nu \det\left| I_\nu\!\left(\frac{2|\rho|\sqrt{\beta_k \alpha_l}}{1-|\rho|^2}\right)\right|}{m! m! \prod_{j=0}^{m-1} j!(j+\nu)! |\rho|^{mn-m} (1-|\rho|^2)^m}, \tag{3.12}$$

[illegible] $m$ [illegible] $\triangle(\boldsymbol{\beta}) = \det\left|\beta_k^{l-1}\right| = \prod_{k>l}(\beta_k - \beta_l)$ [illegible] $\triangle(\boldsymbol{\alpha}) = \det\left|\alpha_k^{l-1}\right| = \prod_{k>l}(\alpha_k - \alpha_l)$

[illegible]. It is straightforward to obtain (3.12) from (3.8) by $2m$ one-to-one nonlinear mappings. $\square$

**4. Joint marginal PDF.** In this section, with $\beta = s^2$ and $\alpha = r^2$, we calculate the joint marginal PDF of $\beta$ and $\alpha$, $p(\beta,\alpha)$, using the techniques and results presented in [4,13]. Then the joint PDF of $s$ and $r$, $p(s,r)$, is easily derived.

If the polynomials $P_k(\beta)$ and $Q_l(\alpha)$ satisfy $\int w(\beta,\alpha) P_k(\beta) Q_l(\alpha) d\beta d\alpha = \delta_{kl}$, then we call $P_k(\beta)$ and $Q_l(\alpha)$ biorthogonal polynomials, associated with the weight function $w(\beta,\alpha)$ [12]. With this definition, we have the following lemma.

LEMMA 4.1. [illegible] $P_k(\beta)$ [illegible] $Q_l(\alpha)$ [illegible] $w(\beta,\alpha)$ [illegible] (3.12) [illegible]

$$p(\boldsymbol{\beta},\boldsymbol{\alpha}) = C_1 \det|P_{k-1}(\beta_l)| \det|w(\beta_k,\alpha_l)| \det|Q_{k-1}(\alpha_l)|, \tag{4.1}$$

[illegible] $C_1$ [illegible]

In this paper, $\nu$ is a nonnegative integer. Using the Hille–Hardy formula [2, p. 185, equation (46)]

$$\sum_{k=0}^{\infty}\frac{k!z^k}{(k+\nu)!}L_k^{\nu}(x)L_k^{\nu}(y)=\frac{(xyz)^{-\frac{\nu}{2}}}{1-z}\exp\left(-z\frac{x+y}{1-z}\right)I_{\nu}\left(\frac{2\sqrt{xyz}}{1-z}\right),|z|<1, \tag{4.2}$$

with $L_k^{\nu}(x)=\frac{1}{k!}e^{x}x^{-\nu}\frac{d^k}{dx^k}(e^{-x}x^{k+\nu})$ as the associated Laguerre polynomial, we can rewrite (3.12) as

$$p(\boldsymbol{\beta},\boldsymbol{\alpha})=\frac{\triangle(\boldsymbol{\beta})\triangle(\boldsymbol{\alpha})\det\left|\beta_k^{\nu}e^{-\beta_k}\alpha_l^{\nu}e^{-\alpha_l}\sum_{j=0}^{\infty}\frac{j!|\rho|^{2j}L_j^{\nu}(\beta_k)L_j^{\nu}(\alpha_l)}{(j+\nu)!}\right|}{m!m!\prod_{j=0}^{m-1}j!(j+\nu)!|\rho|^{m(m-1)}}. \tag{4.3}$$

We set the weight function $w(\beta,\alpha)$ as

$$\begin{aligned} w(\beta,\alpha)&=\beta^{\nu}\alpha^{\nu}e^{-(\beta+\alpha)}\sum_{j=0}^{\infty}\frac{j!|\rho|^{2j}L_j^{\nu}(\beta)L_j^{\nu}(\alpha)}{(j+\nu)!}\\ &=\frac{(\beta\alpha)^{\frac{\nu}{2}}e^{-\frac{\beta+\alpha}{1-|\rho|^2}}I_{\nu}\left(\frac{2|\rho|\sqrt{\beta\alpha}}{1-|\rho|^2}\right)}{(1-|\rho|^2)|\rho|^{\nu}}. \end{aligned} \tag{4.4}$$

It is easy to check that the corresponding biorthogonal polynomials are given by

$$P_k(\beta)=\sqrt{\frac{k!}{(k+\nu)!}}|\rho|^{-k}L_k^{\nu}(\beta), \tag{4.5}$$

$$Q_l(\alpha)=\sqrt{\frac{l!}{(l+\nu)!}}|\rho|^{-l}L_l^{\nu}(\alpha), \tag{4.6}$$

using the following integral equality [2, p. 267, equation 7.414.3]

$$\int_0^{\infty}e^{-x}x^{\nu}L_k^{\nu}(x)L_l^{\nu}(x)=\frac{(k+\nu)!}{k!}\delta_{kl}. \tag{4.7}$$

Moreover, by the addition of multiples of rows of lower order which do not change the determinant of the Vandermonde matrix, each of the rows can be expressed in terms of orthogonal polynomials with respect to the weight function $w(\beta,\alpha)$. Therefore two $m$-dimensional Vandermonde determinants, $\triangle(\boldsymbol{\beta})$ and $\triangle(\boldsymbol{\alpha})$, can be represented as

$$\triangle(\boldsymbol{\beta})=\det\left|\beta_k^{l-1}\right|=C_2\det\left|P_{k-1}(\beta_l)\right|, \tag{4.8}$$

$$\triangle(\boldsymbol{\alpha})=\det\left|\alpha_k^{l-1}\right|=C_3\det\left|Q_{k-1}(\alpha_l)\right|, \tag{4.9}$$

where we use the fact that the matrix transpose does not change the determinant, i.e., $\det|P_{l-1}(\beta_k)|=\det|P_{k-1}(\beta_l)|$ and $\det|Q_{l-1}(\beta_k)|=\det|Q_{k-1}(\beta_l)|$.

The coefficient of $x^k$ in $L_k^{\nu}(x)$ is $\frac{(-1)^k}{k!}$, and the coefficient of $x^k$ in $P_k(x)$ is $(-1)^k|\rho|^{-k}\frac{1}{\sqrt{k!(k+\nu)!}}$; therefore, we have $C_2=\prod_{j=0}^{m-1}(-1)^j|\rho|^j\sqrt{j!(j+\nu)!}=(-1)^{\frac{m(m-1)}{2}}$ $\times\sqrt{|\rho|^{m(m-1)}\prod_{j=0}^{m-1}j!(j+\nu)!}$, obtained by plugging (4.5) into (4.8). Similarly, substitution of (4.6) into (4.9) gives $C_3=C_2$. Now the product of (4.8) and (4.9) results in

$$\triangle(\boldsymbol{\beta})\triangle(\boldsymbol{\alpha}) = |\rho|^{m(m-1)} \prod_{j=0}^{m-1} j!(j+\nu)! \det|P_{k-1}(\beta_l)| \det|Q_{k-1}(\alpha_l)|. \tag{4.10}$$

Based on (4.4) and (4.10), one can see that (4.3) is equal to (4.1) with $C_1 = \frac{1}{m!m!}$. ☐

THEOREM 4.2. [illegible] $\beta$ [illegible] $\alpha$ [illegible]

$$\begin{aligned} p(\beta,\alpha) = {} & \frac{(\beta\alpha)^{\frac{\nu}{2}} e^{-\frac{\beta+\alpha}{1-|\rho|^2}} I_\nu\!\left(\frac{2|\rho|\sqrt{\beta\alpha}}{1-|\rho|^2}\right)}{m^2(1-|\rho|^2)|\rho|^\nu} \sum_{k=0}^{m-1} \frac{k!}{(k+\nu)!} \frac{L_k^\nu(\beta)L_k^\nu(\alpha)}{|\rho|^{2k}} \\ & + \frac{(\beta\alpha)^\nu e^{-(\beta+\alpha)}}{m^2} \sum_{0\le k<l}^{m-1} \left\{ \frac{k!l!}{(k+\nu)!(l+\nu)!} \Big\{ \left[L_k^\nu(\beta)L_l^\nu(\alpha)\right]^2 + \left[L_l^\nu(\beta)L_k^\nu(\alpha)\right]^2 \right. \\ & \left. - \left[|\rho|^{2(l-k)} + |\rho|^{2(k-l)}\right] L_k^\nu(\beta)L_l^\nu(\beta)L_k^\nu(\alpha)L_l^\nu(\alpha) \Big\} \right\}. \end{aligned} \tag{4.11}$$

[illegible]. Based on Lemma 4.1, and the results presented in [13, equation (3.7)] [4], $p(\beta,\alpha)$ can be expressed as

$$m^2\, p(\beta,\alpha) = w(\beta,\alpha) \sum_{k=0}^{m-1} P_k(\beta)Q_k(\alpha) + \sum_{0\le k<l}^{m-1} \det\begin{vmatrix} P_k(\beta) & \overline{P}_k(\alpha) \\ P_l(\beta) & \overline{P}_l(\alpha) \end{vmatrix} \det\begin{vmatrix} \overline{Q}_k(\beta) & Q_k(\alpha) \\ \overline{Q}_l(\beta) & Q_l(\alpha) \end{vmatrix}, \tag{4.12}$$

where $P_k(x)$ and $Q_k(x)$ are defined in (4.5) and (4.6), respectively, the weight function is presented in (4.4), and $\overline{P}_k(\alpha)$ and $\overline{Q}_l(\beta)$ are similarly defined as [13]

$$\overline{P}_k(\alpha) = \int P_k(\beta)w(\beta,\alpha)d\beta = \sqrt{\frac{k!}{(k+\nu)!}}\alpha^\nu e^{-\alpha}|\rho|^k L_k^\nu(\alpha), \tag{4.13}$$

$$\overline{Q}_l(\beta) = \int Q_l(\alpha)w(\beta,\alpha)d\alpha = \sqrt{\frac{l!}{(l+\nu)!}}\beta^\nu e^{-\beta}|\rho|^l L_l^\nu(\beta). \tag{4.14}$$

Plugging (4.4), (4.5), (4.6), (4.13), and (4.14) into (4.12), we arrive at (4.11). ☐

It is straightforward to obtain the joint PDF of $s$ and $r$ from (4.11), according to these one-to-one mappings $s = \sqrt{\beta}$ and $r = \sqrt{\alpha}$.

The joint PDF in (4.11) includes many existing PDFs as special cases.

- By integration over $\beta$, (4.11) reduces to the marginal PDF

  $$p(\alpha) = \frac{1}{m} \sum_{k=0}^{m-1} \frac{k!}{(k+\nu)!} \left[L_k^\nu(\alpha)\right]^2 \alpha^\nu e^{-\alpha}, \tag{4.15}$$

  which is the same as the PDF presented in [18]. When $m = 1$, (4.15) further reduces to

  $$p(\alpha) = \frac{1}{(n-1)!}\alpha^{n-1}e^{-\alpha}, \tag{4.16}$$

  which is the $\chi^2$ distribution with $2n$ degrees of freedom [17, equation (2.32)].

- With $m = 1$, (4.11) reduces to [21]

$$p(\alpha,\beta) = \frac{(\alpha\beta)^{\frac{n-1}{2}} \exp\left(-\frac{\alpha+\beta}{1-|\rho|^2}\right) I_{n-1}\left(\frac{2|\rho|\sqrt{\alpha\beta}}{1-|\rho|^2}\right)}{(n-1)!\,(1-|\rho|^2)\,|\rho|^{n-1}}. \tag{4.17}$$

Furthermore, when $n = 1$, (4.17) simplifies to

$$p(\alpha,\beta) = \frac{1}{1-|\rho|^2} \exp\left(-\frac{\alpha+\beta}{1-|\rho|^2}\right) I_0\left(\frac{2|\rho|\sqrt{\alpha\beta}}{1-|\rho|^2}\right), \tag{4.18}$$

which is identical to [3, p. 163, equation (8-103)], after two one-to-one nonlinear transformations.

For the application discussed in section 5, we need the joint marginal PDF of $\phi$ and $\varphi$, $p(\phi,\varphi)$, where $\phi$ and $\varphi$ are randomly selected from $\{\alpha_k\}_{k=1}^m$, $m \geq 2$. Using the technique in [4, 13], we have the following theorem.

THEOREM 4.3. [illegible] $\phi$ [illegible] $\varphi$ [illegible] $\{\alpha_k\}_{k=1}^m$ [illegible]

$$p(\phi,\varphi) = \frac{(\phi\varphi)^\nu e^{-(\phi+\varphi)}}{m(m-1)} \sum_{\substack{k,l=0\\k\neq l}}^{m-1} \frac{k!\,l!}{(k+\nu)!(l+\nu)!} \left\{\left[L_k^\nu(\phi)L_l^\nu(\varphi)\right]^2 - L_k^\nu(\phi)L_l^\nu(\phi)L_k^\nu(\varphi)L_l^\nu(\varphi)\right\}. \tag{4.19}$$

[illegible] According to (1.6) and (2.14) in [13] we have

$$p(\phi,\varphi) = \frac{1}{m(m-1)} \det\begin{vmatrix} K(\phi,\phi) & K(\phi,\varphi) \\ K(\varphi,\phi) & K(\varphi,\varphi) \end{vmatrix}, \tag{4.20}$$

where $K(x_1,x_2) = \sum_{k=0}^{m-1} P_k(x_1)\overline{Q}_k(x_2)$. With $P_k(x_1)$ in (4.5) and $\overline{Q}_k(x_2)$ in (4.14), we obtain (4.19) after some simple algebraic manipulations. □

## 5. Application to wireless MIMO communication systems.

For an $N_R \times N_T$ MIMO time-varying Rayleigh flat fading channel [19] with $N_T$ transmitters and $N_R$ receivers, the channel impulse response at time instant $t$ is given by

$$\mathbf{H}(t) = \begin{bmatrix} h_{1,1}(t) & \cdots & h_{1,N_T}(t) \\ \vdots & \ddots & \vdots \\ h_{N_R,1}(t) & \cdots & h_{N_R,N_T}(t) \end{bmatrix}. \tag{5.1}$$

We assume all of the $N_R N_T$ subchannels in the MIMO system $\{h_{i,j}(t)\}_{(i=1,j=1)}^{(N_R,N_T)}$ are i.i.d., with the same temporal correlation coefficient, i.e.,

$$\mathbb{E}[h_{ij}(t)h_{pq}^\star(t-\tau)] = \delta_{ip}\delta_{jq}\rho_h(\tau), \tag{5.2}$$

where $\rho_h(\tau) = J_0(2\pi f_D \tau)$ [10] in isotropic scattering environments,[1] with $J_0(x) = I_0(-\jmath x)$ [5, p. 961, equation 8.406.3], and $f_D$ is the maximum Doppler frequency shift.

---

[1] In the nonisotropic scattering environment, $\rho_h(\tau)$, in general, is a complex-value function [22,23], and $|\rho_h(\tau)|$ indicates its amplitude at the time delay $\tau$.

We set $n = \max(N_R, N_T)$ and $m = \min(N_R, N_T)$. According to (2.2), $\mathbf{H}(t)$ can be diagonalized as

$$\mathbf{H}(t) = \mathbf{U}(t)\mathbf{S}(t)\mathbf{V}^{\dagger}(t), \tag{5.3}$$

where $\mathbf{S}(t) = \left[\operatorname{diag}(\mathbf{s}(t))\ \mathbf{0}\right]$, with $\mathbf{s}(t) = [s_1(t), s_2(t), \ldots, s_m(t)]$, for $N_R \leq N_T$, and $\mathbf{S}(t) = \begin{bmatrix} \operatorname{diag}(\mathbf{s}(t)) \\ \mathbf{0} \end{bmatrix}$ for $N_R > N_T$. Therefore the MIMO channel $\mathbf{H}(t)$ is decomposed to $m$ identically distributed eigenchannels $\lambda_k(t) = s_k^2(t)$, $k \in [1, m]$, by SVD.

In wireless MIMO communication systems, we are interested in the correlation coefficient between any two eigenchannels, which is defined by

$$\rho_{k,l}(\tau) = \frac{\mathbb{E}\left[\lambda_k(t)\lambda_l(t-\tau)\right] - \mathbb{E}\left[\lambda_k(t)\right]\mathbb{E}\left[\lambda_l(t)\right]}{\sqrt{\mathbb{E}\left[\lambda_k^2(t)\right] - \left\{\mathbb{E}\left[\lambda_k(t)\right]\right\}^2}\sqrt{\mathbb{E}\left[\lambda_l^2(t)\right] - \left\{\mathbb{E}\left[\lambda_l(t)\right]\right\}^2}}. \tag{5.4}$$

For simplicity, in this paper we consider only a 2×2 MIMO system,[2] $N_R = N_T = 2$, where the correlation coefficient $\rho_{k,l}(\tau)$ can be shown to be

$$\rho_{k,l}(\tau) = \begin{cases} 1 - \frac{3}{2}\left(1 - \delta_{kl}\right), & \tau = 0, \\ \frac{|\rho_h(\tau)|^2}{4} = \frac{J_0^2(2\pi f_D \tau)}{4}, & \tau \neq 0, \end{cases} \quad k, l = 1, 2, \tag{5.5}$$

with $J_0^2(\cdot) = [J_0(\cdot)]^2$. To derive (5.5), we note that, for $\tau = 0$ and $k = l$, $\rho_{k,l}(0) = 1$ because of the definition of the correlation coefficient. Since $m = 2$, for any eigenchannel at the time instant $t$, it is easy to show that the mean value of $\lambda_k(t)$ is $\mathbb{E}\left[\lambda_k(t)\right] = 2\ \forall k$, and the second moment of $\lambda_k(t)$ is $\mathbb{E}\left[\lambda_k^2(t)\right] = 8\ \forall k$, using the PDF in (4.15). For $\tau = 0$ and $k \neq l$, we obtain $\mathbb{E}\left[\lambda_k(t)\lambda_l(t)\right] = 2$ by (4.19); hence, $\rho_{k,l}(0) = -\frac{1}{2}\ \forall k \neq l$. For $\tau \neq 0$ and $\forall k, l$, it is not difficult to get $\mathbb{E}\left[\lambda_k(t)\lambda_l(t-\tau)\right] = 4 + |\rho_h(\tau)|^2$ using (4.11); therefore, we have the second line in (5.5).

Monte Carlo simulations are performed to verify the result in (5.5). In all simulations,[3] the maximum Doppler frequency $f_D$ is set to 1 Hz, and the sampling period $T_s$ is equal to $\frac{1}{1000 f_D}$. The simulation results are shown in Figure 5.1, where the upper figure shows the channel correlation coefficient $\rho_h(\tau) = J_0\left(2\pi f_D \tau\right)$, Clarke's correlation model, whereas the lower figure presents the correlation coefficient between any two eigenchannels or for any individual eigenchannel, (5.5). Since $J_0(2\pi f_D \tau)$ is an even function of $\tau$, the correlation coefficients are plotted for $\tau \geq 0$. In the figure, "Simu." indicates the curve is obtained by Monte Carlo simulations, whereas "Theo." means theoretical. From Figure 5.1 we can conclude that the new theoretical result in (5.5) is confirmed by simulation very well.

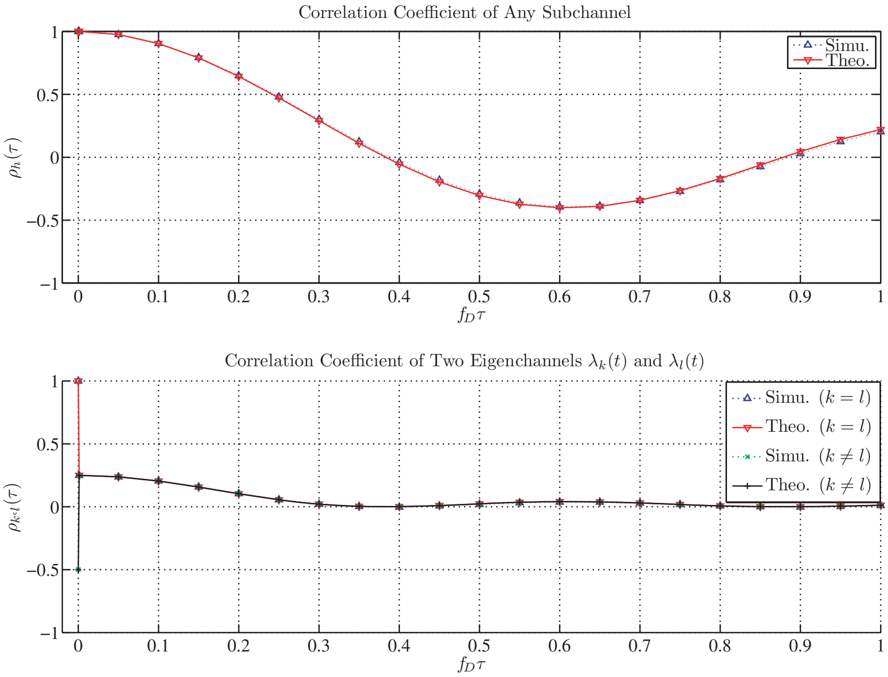
**6. Conclusion.** In this paper, the joint distribution of singular values of two correlated rectangular complex Gaussian random matrices is derived, as well as the joint marginal distribution. The derived distributions play an important role in the analysis and design of wireless MIMO communication systems. As an example, the correlation coefficient of any two eigenchannels of a 2 × 2 MIMO system is obtained and verified by the Monte Carlo simulations in this paper.

**Acknowledgment.** We thank the anonymous reviewers for bringing [6] and [20] to our attention.

---

[2] The general $N_R \times N_T$ MIMO system is considered in [22, Chap. 4].

[3] The spectral method [1] is used to generate the MIMO channels.



Correlation Coefficient of Any Subchannel

Correlation Coefficient of Two Eigenchannels $\lambda_k(t)$ and $\lambda_l(t)$

FIG. 5.1. *The channel correlation coefficient $\rho_h(\tau)$ and correlation coefficient of any two eigenchannels $\rho_{k,l}(\tau)$ in a $2 \times 2$ MIMO system with Clarke's correlation model. Note that the sampling period $T_s$ is $\frac{1}{1000 f_D}$ in Monte Carlo simulations; therefore, the first nonzero $\tau$ is $T_s$, i.e., $\frac{1}{1000 f_D}$, which corresponds to $f_D\tau = \frac{1}{1000}$ in the horizontal axis.*

## REFERENCES

[1] K. ACOLATSE AND A. ABDI, *Efficient simulation of space-time correlated MIMO mobile fading channels*, in Proceedings of the IEEE Vehicle Technology Conference, Orlando, FL, 2003, pp. 652–656.

[2] P. BECKMANN, *Orthogonal Polynomials for Engineers and Physicists*, Golem Press, Boulder, CO, 1973.

[3] W. B. DAVENPORT AND W. L. ROOT, *An Introduction to the Theory of Random Signals and Noise*, Wiley, New York, 1987.

[4] B. EYNARD AND M. L. MEHTA, *Matrices coupled in a chain:* I. *Eigenvalue correlations*, J. Phys. A, 31 (1998), pp. 4449–4456.

[5] I. S. GRADSHTEYN, I. M. RYZHIK, AND A. JEFFREY, EDS., *Table of Integrals, Series, and Products*, 5th ed., Academic, San Diego, CA, 1994.

[6] T. GUHR AND T. WETTIG, *Universal spectral correlations of the Dirac operator at finite temperature*, Nuclear Phys. B, 506 (1997), pp. 589–611.

[7] A. K. GUPTA AND D. K. NAGAR, *Matrix Variate Distributions*, Chapman & Hall/CRC, New York, 1999.

[8] L. K. HUA, *Harmonic Analysis of Functions of Several Complex Variables in the Classical Domain*, American Mathematical Society, Providence, RI, 1963.

[9] A. D. JACKSON, M. K. ŞENER, AND J. J. M. VERBAARSCHOT, *Finite volume partition functions and Itzykson-Zuber integrals*, Phys. Lett. B, 387 (1996), pp. 355–360.

[10] W. C. JAKES, ED., *Microwave Mobile Communications*, IEEE Press, New York, 1994.

[11] A. T. JAMES, *Distributions of matrix variates and latent roots derived from normal samples*, Ann. Math. Statist., 35 (1964), pp. 475–501.

[12] M. L. MEHTA, *Random Matrices*, Academic Press, Boston, MA, 2004.

[13] M. L. MEHTA AND P. SHUKLA, *Two coupled matrices: Eigenvalue correlations and spacing functions*, J. Phys. A, 27 (1994), pp. 7793–7803.

[14] R. J. MUIRHEAD, *Aspects of Multivariate Statistical Theory*, Wiley, New York, 1982.

[15] T. NAGAO AND M. WADATI, *Correlation functions of random matrix ensembles related to classical orthogonal polynomials*, J. Phys. Soc. Japan, 60 (1991), pp. 3298–3322.

[16] J. SHEN, *On the singular values of Gaussian random matrices*, Linear Algebra Appl., 326 (2001), pp. 1–14.

[17] M. K. SIMON, *Probability Distributions Involving Gaussian Random Variables: A Handbook for Engineers and Scientists*, Kluwer, Boston, MA, 2002.

[18] İ. E. TELATAR, *Capacity of multi-antenna Gaussian channels*, European Trans. Telecommun., 10 (1999), pp. 585–595.

[19] D. TSE AND P. VISWANATH, *Fundamentals of Wireless Communication*, Cambridge University Press, Cambridge, 2005.

[20] J. J. M. VERBAARSCHOT AND T. WETTIG, *Random matrix theory and chiral symmetry in QCD*, Annu. Rev. Nuclear Particle Sci., 50 (2000), pp. 343–410.

[21] S. WANG, *Envelope correlation coefficient for logarithmic diversity receivers revisited*, IEEE Trans. Commun., to appear.

[22] S. WANG, *MIMO Fading Channels: Models, Statistics, and Low-Complexity Estimators*, Ph.D. thesis, New Jersey Institute of Technology, Newark, NJ, 2006.

[23] S. WANG AND A. ABDI, *On the second-order statistics of the instantaneous mutual information of time-varying fading channels*, in Proceedings of the IEEE International Workshop Signal Processing Advances in Wireless Communications, New York, 2005, pp. 405–409.

[24] S. WANG AND A. ABDI, *Correlation analysis of instantaneous mutual information in* $2 \times 2$ *MIMO systems*, in Proceedings of the 40th Annual Conference on Information Science Systems, Princeton, NJ, 2006, pp. 542–546.

[25] S. WANG AND A. ABDI, *Statistical characterization of eigen-channels in time-varying Rayleigh flat fading MIMO systems*, in Proceedings of the IEEE Global Telecommunications Conference, San Francisco, CA, 2006.

[26] N. ZHANG AND B. VOJCIC, *Evaluating the temporal correlation of MIMO channel capacities*, in Proceedings of the IEEE Global Telecommunications Conference, St. Louis, MO, 2005, pp. 2817–2821.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 982–1006

# PROJECTED GENERALIZED DISCRETE-TIME PERIODIC LYAPUNOV EQUATIONS AND BALANCED REALIZATION OF PERIODIC DESCRIPTOR SYSTEMS*

ERIC KING-WAH CHU†, HUNG-YUAN FAN‡, AND WEN-WEI LIN§

**Abstract.** From the necessary and sufficient conditions for complete reachability and observability of periodic descriptor systems with time-varying dimensions, the symmetric positive semidefinite reachability/observability Gramians are defined. These Gramians can be shown to satisfy some projected generalized discrete-time periodic Lyapunov equations. We propose a numerical method for solving these projected Lyapunov equations, and give an illustrative numerical example. As an application of our results, the balanced realization of periodic descriptor systems is discussed.

**Key words.** periodic systems, descriptor systems, reachability and observability Gramians, Hankel singular values, balanced realization

**AMS subject classifications.** 15A24, 93B05, 93B07, 93C55

**DOI.** 10.1137/040606715

## 1. Introduction.

In the second half of the last century, the development of systems and control theory, together with the achievements of digital control and signal processing, has set the stage for renewed interest in the study of periodic systems, both in continuous and discrete time; see, e.g., [11, 13, 16, 27, 42, 53] and the survey papers [3, 4]. This has been amplified by specific application demands in the aerospace realm [20, 21, 28], computer control of industrial processes [5], and communication systems [11, 40, 41, 52]. The number of contributions on linear time-varying discrete-time periodic systems has been increasing in recent times; see, e.g., [14, 19, 22, 43, 45, 47] and the references therein. This increasing interest in periodic systems has also been motivated by the large variety of processes that can be modeled through linear discrete-time periodic systems (e.g., multirate sampled-data systems, chemical processes, periodically time-varying filters and networks, and seasonal phenomena [2, 3, 6, 15, 26, 31, 51]).

We consider here a periodic descriptor system with time-varying dimensions:

$$E_k x_{k+1} = A_k x_k + B_k u_k, \quad y_k = C_k x_k, \quad k \in \mathbb{Z}, \tag{1.1}$$

where the matrices $E_k \in \mathbb{R}^{\mu_{k+1}\times n_{k+1}}$, $A_k \in \mathbb{R}^{\mu_{k+1}\times n_k}$, $B_k \in \mathbb{R}^{\mu_{k+1}\times m}$, $C_k \in \mathbb{R}^{p\times n_k}$ are periodic with period $K \geq 1$, i.e., $E_k = E_{k+K}$, $A_k = A_{k+K}$, $B_k = B_{k+K}$, $C_k = C_{k+K}$, and all system matrices $E_k$ and $A_k$ are allowed to be rectangular for all $k$. Moreover, the dimensions of matrices are also $K$-periodic, i.e., $\mu_{k+K} = \mu_k$ and $n_{k+K} = n_k$ for all $k$. Assume that $\sum_{k=0}^{K-1} \mu_k = \sum_{k=0}^{K-1} n_k \equiv N$. Recently, this class of periodic descriptor systems (1.1) has been discussed and studied extensively in the problem of solvability

---

*Received by the editors April 14, 2004; accepted for publication (in revised form) by P. Van Dooren February 20, 2007; published electronically October 17, 2007.
http://www.siam.org/journals/simax/29-3/60671.html

†School of Mathematical Sciences, Building 28, Monash University, VIC 3800, Australia (eric.chu@sci.monash.edu.au).

‡Department of Applied Mathematics, Hsuan Chuang University, Hsinchu, 300, Taiwan (hyfan@wmail.hcu.edu.tw). This author's research was partly supported by the National Science Council in Taiwan under grant NSC 94-2115-M-264-003.

§Department of Mathematics, National Tsing Hua University, Hsinchu, 300, Taiwan (wwlin@math.nthu.edu.tw).

and conditionability [33], the computation of $H_\infty$-norm and system zeros [49, 50], and the compensating and regularization problems for periodic descriptor systems [8, 23].

It is well known that the dynamics of the discrete-time periodic descriptor system (1.1) depend critically on the regularity and the eigenstructure of the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$, which satisfy the homogeneous systems of (1.1):

$$E_k x_{k+1} = A_k x_k, \quad k \in \mathbb{Z}. \tag{1.2}$$

The set of matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ is said to be regular when $\det[C((\alpha_k, \beta_k)_{k=0}^{K-1})] \neq 0$, where

$$C((\alpha_k, \beta_k)_{k=0}^{K-1}) \equiv \begin{bmatrix} \alpha_0 E_0 & 0 & \cdots & 0 & -\beta_0 A_0 \\ -\beta_1 A_1 & \alpha_1 E_1 & & & 0 \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & 0 \\ 0 & & 0 & -\beta_{K-1} A_{K-1} & \alpha_{K-1} E_{K-1} \end{bmatrix} \in \mathbb{R}^{N \times N}, \tag{1.3}$$

in which $\alpha_k, \beta_k$ are complex variables for $k = 0, \dots, K-1$. Note that we are considering the regularity of the [illegible] of matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$, rather than the regularity of the individual matrix pairs $(E_k, A_k)$.

Definition 1.1. [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] $\alpha_0, \dots, \alpha_{K-1}$ $\beta_0, \dots, \beta_{K-1}$ [illegible]

$$\det[C((\alpha_k, \beta_k)_{k=0}^{K-1})] = 0, \qquad \left( \prod_{k=0}^{K-1} \alpha_k, \prod_{k=0}^{K-1} \beta_k \right) \equiv (\pi_\alpha, \pi_\beta) \neq (0, 0), \tag{1.4}$$

[illegible] $(\pi_\alpha, \pi_\beta)$ [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$

Note that if $(\pi_\alpha, \pi_\beta)$ is an eigenvalue pair of $\{(E_k, A_k)\}_{k=0}^{K-1}$, then $(\pi_\alpha, \pi_\beta)$ and $(\tau\pi_\alpha, \tau\pi_\beta)$ represent the same eigenvalue for any nonzero $\tau$. If $\pi_\beta \neq 0$, then $\lambda = \pi_\alpha/\pi_\beta$ is a finite eigenvalue; otherwise $(\pi_\alpha, 0)$ represents an infinite eigenvalue. We shall assume throughout the paper that the set of periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ is regular, and use the notation $\sigma(M)$ to denote the spectrum of a square matrix $M$.

For discrete-time descriptor systems, the concepts of reachability and observability Gramians, causal and noncausal Hankel singular values, and balanced realization are well established [1, 39]. Moreover, numerical methods have been proposed in [10, 34] to solve the projected generalized Lyapunov equations for continuous-time descriptor systems. However, to the best of our knowledge, similar results have not been developed for periodic descriptor systems.

In summary, there are two main contributions in this paper. First, with the aid of the fundamental matrices $\Psi_{i,j}$ defined as in (2.15), the reachability/observability Gramians and their corresponding projected generalized discrete-time periodic Lyapunov equations (GDPLEs) are derived in terms of the original system matrices $E_k$, $A_k$, $B_k$, and $C_k$, $k = 0, 1, \dots, K-1$, respectively. These fundamental matrices play an important role here and are natural extensions of those defined for the descriptor system with period $K = 1$ [34, 39]. Second, in sections 6 and 7, Hankel singular values and balanced realization are discussed, for the first time, for completely reachable and observable periodic descriptor systems. These concepts are likely to be crucial in the model reduction problem of periodic descriptor systems.

This paper is organized as follows. Section 2 contains some notation and definitions, as well as some preliminary results. In section 3 the necessary and sufficient conditions for complete reachability and observability of periodic descriptor systems are developed from similar results for systems with constant dimensions in [10]. With these equivalent conditions, the periodic reachability and observability Gramians, which satisfy some generalized periodic Lyapunov equations, are developed in section 4. In section 5 we propose a numerical method for solving these equations under the assumption of pd-stability. A numerical example is given to illustrate its feasibility and reliability. The concept of Hankel singular values is generalized to periodic descriptor systems in section 6. The problem of balanced realization for the completely reachable and completely observable periodic descriptor systems is discussed in section 7.

**2. Preliminaries.** For period $K = 1$ and a regular matrix pair $(E, A)$, it is well known that the discrete-time descriptor system $(E, A, B, C)$ is asymptotically stable if and only if all finite eigenvalues of $(E, A)$ lie inside the unit circle [12, 35, 37]. Similarly, the stability of the periodic descriptor system (1.1) can be characterized in terms of the spectrum of the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$.

DEFINITION 2.1. [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

In a fashion similar to the Kronecker canonical form for a regular matrix pair, we can transform a regular set of periodic matrix pairs into a periodic Kronecker canonical form [48] (see also [32] for the history of the canonical form).

LEMMA 2.1. [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] (1.1) [illegible] $k = 0, \ldots, K-1$ [illegible] $X_k \in \mathbb{R}^{\mu_{k+1} \times \mu_{k+1}}$ [illegible] $Y_k \in \mathbb{R}^{n_k \times n_k}$ [illegible]

$$X_k E_k Y_{k+1} = \begin{bmatrix} I_{n_{k+1}^f} & 0 \\ 0 & E_k^b \end{bmatrix}, \quad X_k A_k Y_k = \begin{bmatrix} A_k^f & 0 \\ 0 & I_{n_k^\infty} \end{bmatrix}, \tag{2.1}$$

[illegible] $Y_K \equiv Y_0$ $A_{k+K-1}^f A_{k+K-2}^f \cdots A_k^f \equiv J_k$ [illegible] $n_k^f \times n_k^f$ [illegible] $E_k^b E_{k+1}^b \cdots E_{k+K-1}^b \equiv N_k$ [illegible] $n_k^\infty \times n_k^\infty$ [illegible] $n_k = n_k^f + n_k^\infty$ [illegible] $\mu_{k+1} = n_{k+1}^f + n_k^\infty$

[illegible] Since $\{(E_k, A_k)\}_{k=0}^{K-1}$ are pd-stable, there exist orthogonal matrices $V_k \in \mathbb{R}^{\mu_{k+1} \times \mu_{k+1}}$ and $U_k \in \mathbb{R}^{n_k \times n_k}$, with $U_K \equiv U_0$ and for $k = 0, 1, \ldots, K-1$, such that

$$V_k^T E_k U_{k+1} = \begin{bmatrix} E_{k,1} & E_{k,3} \\ 0 & E_{k,2} \end{bmatrix}, \quad V_k^T A_k U_k = \begin{bmatrix} A_{k,1} & A_{k,3} \\ 0 & A_{k,2} \end{bmatrix}, \tag{2.2}$$

where the matrices $E_{k,1} \in \mathbb{R}^{n_{k+1}^f \times n_{k+1}^f}$ and $A_{k,2} \in \mathbb{R}^{n_k^\infty \times n_k^\infty}$ are nonsingular and

$$(A_{k,2})^{-1} E_{k,2} (A_{k+1,2})^{-1} E_{k+1,2} \cdots (A_{k+K-1,2})^{-1} E_{k+K-1,2}$$

are nilpotent for $k = 0, 1, \ldots, K-1$ [48]. All finite eigenvalues of the periodic matrix pairs $\{(E_{k,1}, A_{k,1})\}_{k=0}^{K-1}$ lie inside the unit circle, and the spectrum of the periodic matrix pairs $\{(E_{k,2}, A_{k,2})\}_{k=0}^{K-1}$ contains only infinite eigenvalues. We then construct

$$\begin{bmatrix} E_{k,1}^{-1} & 0 \\ 0 & A_{k,2}^{-1} \end{bmatrix} \begin{bmatrix} E_{k,1} & E_{k,3} \\ 0 & E_{k,2} \end{bmatrix} = \begin{bmatrix} I_{n_{k+1}^f} & \hat{E}_{k,3} \\ 0 & \hat{E}_k^\infty \end{bmatrix}$$

and

$$\begin{bmatrix} E_{k,1}^{-1} & 0 \\ 0 & A_{k,2}^{-1} \end{bmatrix} \begin{bmatrix} A_{k,1} & A_{k,3} \\ 0 & A_{k,2} \end{bmatrix} = \begin{bmatrix} \hat{A}_k^f & \hat{A}_{k,3} \\ 0 & I_{n_k^\infty} \end{bmatrix},$$

where $\hat{E}_{k,3} \in \mathbb{R}^{n_{k+1}^f \times n_{k+1}^\infty}$, $\hat{E}_k^\infty \in \mathbb{R}^{n_k^\infty \times n_{k+1}^\infty}$, $\hat{A}_k^f \in \mathbb{R}^{n_{k+1}^f \times n_k^f}$, and $\hat{A}_{k,3} \in \mathbb{R}^{n_{k+1}^f \times n_k^\infty}$.

We shall prove that there exist periodic matrices $P_k \in \mathbb{R}^{n_{k+1}^f \times n_k^\infty}$ and $Q_k \in \mathbb{R}^{n_k^f \times n_k^\infty}$ such that

$$\begin{bmatrix} I_{n_{k+1}^f} & P_k \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} I_{n_{k+1}^f} & \hat{E}_{k,3} \\ 0 & \hat{E}_k^\infty \end{bmatrix} \begin{bmatrix} I_{n_{k+1}^f} & Q_{k+1} \\ 0 & I_{n_{k+1}^\infty} \end{bmatrix} = \begin{bmatrix} I_{n_{k+1}^f} & 0 \\ 0 & \hat{E}_k^\infty \end{bmatrix} \tag{2.3}$$

and

$$\begin{bmatrix} I_{n_{k+1}^f} & P_k \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} \hat{A}_k^f & \hat{A}_{k,3} \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} I_{n_k^f} & Q_k \\ 0 & I_{n_k^\infty} \end{bmatrix} = \begin{bmatrix} \hat{A}_k^f & 0 \\ 0 & I_{n_k^\infty} \end{bmatrix}. \tag{2.4}$$

Comparing both sides of (2.3) and (2.4), we obtain, for all $k$,

$$Q_{k+1} + P_k \hat{E}_k^\infty + \hat{E}_{k,3} = 0 \tag{2.5}$$

and

$$\hat{A}_k^f Q_k + P_k + \hat{A}_{k,3} = 0. \tag{2.6}$$

Eliminating $P_k$ from (2.5) and (2.6), we arrive at, for all $k$,

$$Q_{k+1} = \hat{A}_k^f Q_k \hat{E}_k^\infty + \hat{A}_{k,3} \hat{E}_k^\infty - \hat{E}_{k,3}. \tag{2.7}$$

With $Q_K = Q_0$, (2.7) in turn implies

$$Q_0 = (\hat{A}_{K-1}^f \hat{A}_{K-2}^f \cdots \hat{A}_0^f) Q_0 (\hat{E}_0^\infty \hat{E}_1^\infty \cdots \hat{E}_{K-1}^\infty) + D_0, \tag{2.8}$$

where $D_0$ is independent of any $Q_k$. Since $\sigma(\hat{E}_0^\infty \hat{E}_1^\infty \cdots \hat{E}_{K-1}^\infty) = \{0\}$, we can uniquely determine $Q_0$ from (2.8), all the other $Q_k$ from (2.7), and all the $P_k$ from (2.6).

Finally, by the well-known Jordan decomposition, there exist nonsingular $K$-periodic matrices $G_k \in \mathbb{R}^{n_k^f \times n_k^f}$ and $Z_k \in \mathbb{R}^{n_k^\infty \times n_k^\infty}$ which produce the Jordan forms

$$J_k \equiv G_k^{-1} (\hat{A}_{k+K-1}^f \hat{A}_{k+K-2}^f \cdots \hat{A}_k^f) G_k, \tag{2.9}$$

$$N_k \equiv Z_k^{-1} (\hat{E}_k^\infty \hat{E}_{k+1}^\infty \cdots \hat{E}_{k+K-1}^\infty) Z_k. \tag{2.10}$$

Define

$$X_k \equiv \begin{bmatrix} G_{k+1}^{-1} & 0 \\ 0 & Z_k^{-1} \end{bmatrix} \begin{bmatrix} I_{n_{k+1}^f} & P_k \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} E_{k,1}^{-1} & 0 \\ 0 & A_{k,2}^{-1} \end{bmatrix} V_k^T,$$

$$Y_k \equiv U_k \begin{bmatrix} I_{n_k^f} & Q_k \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} G_k & 0 \\ 0 & Z_k \end{bmatrix}$$

and

$$E_k^b \equiv Z_k^{-1} \hat{E}_k^\infty Z_{k+1}, \quad A_k^f \equiv G_{k+1}^{-1} \hat{A}_k^f G_k;$$

then the proof is complete. □

[illegible]. (i) If $\nu_k$ is the nilpotency of the matrix $N_k$ for $k = 0, 1, \ldots, K-1$, then these $K$ values are defined as the indices of a regular set of periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ [23]. Hence we define the index of the periodic descriptor system (1.1) as $\nu \equiv \max\{\nu_0, \nu_1, \ldots, \nu_{K-1}\}$. We say that the periodic descriptor system (1.1) is of index at most 1 if $\nu \leq 1$, i.e., $E_k$ are all nonsingular or $N_k = 0$ for all $k$.

(ii) The technique in proving the uniqueness of the solution for (2.5) and (2.6) can be used later for similar equations in (4.7), (4.10), (5.3), (5.6), (5.11), (5.15), (5.17), and (5.21). For the special case of periodic Lyapunov equations for systems with constant dimensions, see [44].

For each $k \in \mathbb{Z}$, we let

$$x_k = Y_k \begin{bmatrix} x_k^f \\ x_k^b \end{bmatrix}\begin{matrix} \}n_k^f \\ \}n_k^\infty \end{matrix}, \quad X_k B_k = \begin{bmatrix} B_k^f \\ B_k^b \end{bmatrix}\begin{matrix} \}n_{k+1}^f \\ \}n_k^\infty \end{matrix}, \quad C_k Y_k = \begin{array}{c} \begin{bmatrix} C_k^f & C_k^b \end{bmatrix} \\ \begin{matrix} n_k^f & n_k^\infty \end{matrix} \end{array}, \tag{2.11}$$

and by using Lemma 2.1, we can decompose the original system (1.1) into forward and backward periodic subsystems, respectively:

$$x_{k+1}^f = A_k^f x_k^f + B_k^f u_k, \quad y_k^f = C_k^f x_k^f, \tag{2.12}$$

$$E_k^b x_{k+1}^b = x_k^b + B_k^b u_k, \quad y_k^b = C_k^b x_k^b, \tag{2.13}$$

with $y_k = y_k^f + y_k^b$, $k \in \mathbb{Z}$.

Notice that the state transition matrix of the forward subsystem (2.12) equals $\Phi_f(i,j) = A_{i-1}^f A_{i-2}^f \cdots A_j^f \in \mathbb{R}^{n_i^f \times n_j^f}$ when $i > j$ with $\Phi_f(i,i) := I_{n_i^f}$. The state transition matrix of the backward subsystem (2.13) is $\Phi_b(i,j) = E_i^b E_{i+1}^b \cdots E_{j-1}^b \in \mathbb{R}^{n_i^\infty \times n_j^\infty}$ when $i < j$ with $\Phi_b(i,i) := I_{n_i^\infty}$. The state transition matrix over one period $\Phi_f(\tau + K, \tau)$ is called the monodromy matrix of the forward subsystem (2.12) at time $\tau$.

For $k = 0, 1, \ldots, K-1$, the matrices

$$P_r(k) = Y_k \begin{bmatrix} I_{n_k^f} & 0 \\ 0 & 0 \end{bmatrix} Y_k^{-1} \in \mathbb{R}^{n_k \times n_k}, \qquad P_l(k) = X_k^{-1} \begin{bmatrix} I_{n_{k+1}^f} & 0 \\ 0 & 0 \end{bmatrix} X_k \in \mathbb{R}^{\mu_{k+1} \times \mu_{k+1}}, \tag{2.14}$$

are the spectral projections onto, respectively, the $k$th right and left deflating subspaces of the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ corresponding to the finite eigenvalues. Moreover, the fundamental matrices $\Psi_{i,j}$ $(i, j \in \mathbb{Z})$ of the periodic descriptor system (1.1) are defined by

$$\Psi_{i,j} = \begin{cases} Y_i \begin{bmatrix} \Phi_f(i, j+1) & 0 \\ 0 & 0 \end{bmatrix} X_j & \text{if } i > j, \\ Y_i \begin{bmatrix} 0 & 0 \\ 0 & -\Phi_b(i,j) \end{bmatrix} X_j & \text{if } i \leq j. \end{cases} \tag{2.15}$$

These matrices play an essential role for the periodic discrete-time descriptor system (1.1). For the discrete-time descriptor system with period $K = 1$, these fundamental matrices coincide with the coefficient matrices of the Laurent expansion of the generalized resolvent $(\lambda E - A)^{-1}$ at infinity [25, 39].

Some of the results in this paper can be developed through the application of the results of Stykel [37], after "lifting" the periodic system into a descriptor system of higher dimension [50]. This is implicitly how we obtained the results in Theorem 4.1. However, various permutations will be involved when separating the lifted system into its forward and backward parts, diluting the simplicity we hope for. Our present approach of development without lifting is somewhat simpler and more convenient.

**3. Complete reachability and observability.** In this section, we shall give a characterization of complete reachability and observability for the periodic discrete-time descriptor systems (1.1). Proofs in [10], for similar results for systems of constant dimensions, can easily be adapted for time-variant dimensions and are omitted.

DEFINITION 3.1. (i) [illegible] (1.1) [illegible] $t$ [illegible] $\bar{x} \in \mathbb{R}^{n_t}$ [illegible] $s, \ell$ [illegible] $s < t < \ell$ [illegible] $\{u_i\}_{i=s}^{\ell}$ [illegible] $x_s = 0$ [illegible] $x_t = \bar{x}$ [illegible] (1.1) [illegible] $t$

(ii) [illegible] (2.12) [illegible] $t$ [illegible] $\bar{\xi}_1 \in \mathbb{R}^{n_t^f}$ [illegible] $s$ [illegible] $s < t$ [illegible] $\{u_i\}_{i=s}^{t-1}$ [illegible] $x_s^f = 0$ [illegible] $x_t^f = \bar{\xi}_1$ [illegible] (2.12) [illegible] $t$

(iii) [illegible] (2.13) [illegible] $t$ [illegible] $\bar{\xi}_2 \in \mathbb{R}^{n_t^\infty}$ [illegible] $\ell$ [illegible] $\ell > t$ [illegible] $\{u_i\}_{i=t}^{\ell}$ [illegible] $x_t^b = \bar{\xi}_2$ [illegible] (2.13) [illegible] $t$

[illegible] It is easily seen from Definition 3.1 that the periodic discrete-time descriptor system (1.1) is completely reachable if and only if both its forward and backward subsystems are completely reachable.

THEOREM 3.1 (forward reachability). [illegible]

(a) [illegible] (2.12) [illegible]

(b) [illegible] $t = 0, 1, 2, \ldots, K-1$ [illegible]

$$\mathcal{R}^f(t) \equiv \left[ B_{t-1}^f,\, A_{t-1}^f B_{t-2}^f, \ldots, \Phi_f(t, t - n_t^f K + 1) B_{t-n_t^f K}^f \right]$$

[illegible]

(c) [illegible] $t = 0, 1, 2, \ldots, K-1$ [illegible]

$$\mathcal{B}_t^f \equiv \left[ B_{t-1}^f,\, A_{t-1}^f B_{t-2}^f,\, A_{t-1}^f A_{t-2}^f B_{t-3}^f, \ldots, \Phi_f(t, t-K+1) B_{t-K}^f \right],$$

[illegible]

$$\left[ \mathcal{B}_t^f,\, \Phi_f(t, t-K)\mathcal{B}_t^f,\, (\Phi_f(t,t-K))^2 \mathcal{B}_t^f, \ldots, (\Phi_f(t,t-K))^{n_t^f - 1} \mathcal{B}_t^f \right]$$

[illegible]

(d) [illegible] $\prod_{i=0}^{K-1} \alpha_i \in \sigma(\Phi_f(K,0))$ [illegible] $U^f(\alpha_0, \ldots, \alpha_{K-1}) \equiv$

$$\left[ \begin{array}{ccccc|cccc} \alpha_0 I_{n_1^f} & 0 & \cdots & 0 & -A_0^f & B_0^f & & & \\ -A_1^f & \alpha_1 I_{n_2^f} & \ddots & & 0 & & B_1^f & & \\ 0 & -A_2^f & \ddots & \ddots & \vdots & & & \ddots & \\ \vdots & \ddots & \ddots & \ddots & 0 & & & & \ddots \\ 0 & \cdots & 0 & -A_{K-1}^f & \alpha_{K-1} I_{n_0^f} & & & & B_{K-1}^f \end{array} \right]$$

[illegible]

(e) [illegible] $t = 0, 1, 2, \dots, K-1$

$$y^T\Phi_f(t+K,t) = \lambda y^T,\ y^T\Phi_f(t,j)B^f_{j-1} = 0$$ [illegible] $j = t-K+1, \dots, t-1, t$

[illegible] $y = 0$

THEOREM 3.2 (backward reachability). [illegible]

(a) [illegible] (2.13) [illegible]

(b) [illegible] $t = 0, 1, 2, \dots, K-1$ [illegible]

$$\mathcal{R}^b(t) \equiv \left[B^b_t,\ E^b_tB^b_{t+1},\ \dots,\ \Phi_b(t, t+\nu K-1)B^b_{t+\nu K-1}\right]$$

[illegible]

(c) [illegible] $t = 0, 1, 2, \dots, K-1$ [illegible]

$$\mathcal{B}^b_t \equiv \left[B^b_t,\ E^b_tB^b_{t+1}, \dots, E^b_tE^b_{t+1}, \dots, E^b_{t+K-2}B^b_{t+K-1}\right],$$

[illegible] $[N_t,\ \mathcal{B}^b_t]$ [illegible]

(d) [illegible] $(\mathcal{E}_b, \mathcal{B}_b)$ [illegible]

(3.1)

$$\mathcal{E}_b \equiv \begin{bmatrix} 0 & E^b_0 & & & \\ 0 & 0 & E^b_1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & 0 & & \ddots & E^b_{K-2} \\ E^b_{K-1} & 0 & \cdots & \cdots & 0 \end{bmatrix}, \quad \mathcal{B}_b \equiv \begin{bmatrix} B^b_0 & & & & \\ & B^b_1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & B^b_{K-1} \end{bmatrix}.$$

DEFINITION 3.2. (i) [illegible] (1.1) [illegible] $t$ [illegible] $s, \ell$ [illegible] $s < t < \ell$ [illegible] $t$ [illegible] $\{y_i\}^{\ell}_{i=s}$ [illegible] $\{u_i\}^{\ell}_{i=s}$ [illegible] (1.1) [illegible] $t$

(ii) [illegible] (2.12) [illegible] $t$ [illegible] $\ell$ [illegible] $\ell > t$ [illegible] $t$ [illegible] $\{y_i\}^{\ell}_{i=t}$ [illegible] $\{u_i\}^{\ell}_{i=t}$ [illegible] (2.12) [illegible] $t$

(iii) [illegible] (2.13) [illegible] $t$ [illegible] $s$ [illegible] $s < t$ [illegible] $t$ [illegible] $\{y_i\}^{t}_{i=s}$ [illegible] $\{u_i\}^{t}_{i=s}$ [illegible] (2.13) [illegible] $t$

[illegible] It is easily seen from Definition 3.2 that the periodic discrete-time descriptor system (1.1) is completely observable if and only if both its forward and backward subsystems are completely observable.

THEOREM 3.3 (forward observability). [illegible]

(a) [illegible] (2.12) [illegible]

(b) [illegible] $t = 0, 1, 2, \dots, K-1$ [illegible]

$$\mathcal{O}^f(t) \equiv \begin{bmatrix} C^f_t \\ C^f_{t+1}A^f_t \\ C^f_{t+2}A^f_{t+1}A^f_t \\ \vdots \\ C^f_{t+n^f_tK-1}\Phi_f(t+n^f_tK-1, t) \end{bmatrix}$$

[illegible]

(c) [illegible] $t = 0, 1, 2, \dots, K-1$ [illegible]

$$\mathcal{C}_t^f \equiv \left[(C_t^f)^T,\, (A_t^f)^T (C_{t+1}^f)^T,\, \dots,\, \Phi_f(t+K-1,t)^T (C_{t+K-1}^f)^T\right]^T ,$$

[illegible]

$$\begin{bmatrix} \mathcal{C}_t^f \\ \mathcal{C}_t^f \Phi_f(t+K,t) \\ \mathcal{C}_t^f (\Phi_f(t+K,t))^2 \\ \vdots \\ \mathcal{C}_t^f (\Phi_f(t+K,t))^{n_t^f - 1} \end{bmatrix}$$

(d) [illegible] $\prod_{i=0}^{K-1} \alpha_i \in \sigma(\Phi_f(K,0))$ [illegible]

$$V^f(\alpha_0, \dots, \alpha_{K-1}) \equiv \left[\begin{array}{ccccc} \alpha_0 I_{n_0^f} & 0 & \cdots & 0 & -A_{K-1}^f \\ -A_0^f & \alpha_1 I_{n_1^f} & \ddots & & 0 \\ 0 & -A_1^f & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -A_{K-2}^f & \alpha_{K-1} I_{n_{K-1}^f} \\ \hline C_0^f & & & & \\ & C_1^f & & & \\ & & \ddots & & \\ & & & C_{K-2}^f & \\ & & & & C_{K-1}^f \end{array}\right]$$

(e) [illegible] $t = 0, 1, 2, \dots, K-1$

$$\Phi_f(t+K,t)x = \lambda x \text{ [illegible] } C_i^f \Phi_f(i,t)x = 0 \text{ [illegible] } i = t, t+1, \dots, t+K-1$$

[illegible] $x = 0$

THEOREM 3.4 (backward observability). [illegible]

(a) [illegible] (2.13) [illegible]

(b) [illegible] $t = 0, 1, 2, \dots, K-1$ [illegible]

$$\mathcal{O}^b(t) \equiv \begin{bmatrix} C_t^b \\ C_{t-1}^b E_{t-1}^b \\ C_{t-2}^b E_{t-2}^b E_{t-1}^b \\ \vdots \\ C_{t-\nu K+1}^b \Phi_b(t-\nu K+1,t) \end{bmatrix}$$

(c) [illegible] $t = 0, 1, 2, \dots, K-1$ [illegible]

$$\mathcal{C}_t^b \equiv \left[(C_t^b)^T,\, (E_{t-1}^b)^T (C_{t-1}^b)^T,\, \dots,\, \Phi_b(t-K+1,t)^T (C_{t-K+1}^b)^T\right]^T ,$$

[illegible]

$$\begin{bmatrix} \mathcal{C}_t^b \\ \mathcal{C}_t^b N_t \\ \mathcal{C}_t^b N_t^2 \\ \\ \mathcal{C}_t^b N_t^{\nu-1} \end{bmatrix}$$

[illegible]

(d) [illegible] $(\mathcal{E}_b, \mathcal{C}_b)$ [illegible] $\mathcal{E}_b$ [illegible] (3.1) [illegible] $\mathcal{C}_b \equiv \operatorname{diag}(C_0^b, C_1^b, \ldots, C_{K-1}^b)$

**4. Periodic reachability and observability Gramians.** It is well known that Gramians play an important role in many applications, such as the model reduction problem [17, 29, 54]. In this section, the concepts of reachability and observability Gramians are generalized to periodic discrete-time descriptor systems (1.1).

Consider the causal and noncausal reachability matrices given by

$$\mathcal{R}_+(t) \equiv \left[\Psi_{t,t-1}B_{t-1},\ \Psi_{t,t-2}B_{t-2}, \ldots,\ \Psi_{t,i}B_i, \ldots\right] \quad (t = 0, 1, \ldots, K-1)$$

and

$$\mathcal{R}_-(t) \equiv \left[\Psi_{t,t}B_t,\ \Psi_{t,t+1}B_{t+1}, \ldots,\ \Psi_{t,t+\nu K-1}B_{t+\nu K-1}\right] \quad (t = 0, 1, \ldots, K-1),$$

respectively, with $\Psi_{i,j}$ $(i, j \in \mathbb{Z})$ as defined in (2.15).

DEFINITION 4.1 (reachability Gramians). [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

(i) [illegible] (1.1) [illegible]

$$G_k^{cr} \equiv \mathcal{R}_+(k)\mathcal{R}_+(k)^T = \sum_{i=-\infty}^{k-1} \Psi_{k,i}B_iB_i^T\Psi_{k,i}^T \in \mathbb{R}^{n_k \times n_k}, \quad k = 0, 1, \ldots, K-1.$$

(ii) [illegible] (1.1) [illegible]

$$G_k^{nr} \equiv \mathcal{R}_-(k)\mathcal{R}_-(k)^T = \sum_{i=k}^{k+\nu K-1} \Psi_{k,i}B_iB_i^T\Psi_{k,i}^T \in \mathbb{R}^{n_k \times n_k}, \quad k = 0, 1, \ldots, K-1.$$

(iii) [illegible] (1.1) [illegible]

$$G_k^r \equiv G_k^{cr} + G_k^{nr}, \quad k = 0, 1, \ldots, K-1.$$

The causal and noncausal observability matrices are respectively defined by

$$\mathcal{O}_+(t) \equiv \left[\Psi_{t,t-1}^TC_t^T,\ \Psi_{t+1,t-1}^TC_{t+1}^T, \ldots, \Psi_{i,t-1}^TC_i^T, \ldots\right]^T \quad (t = 0, 1, \ldots, K-1)$$

and

$$\mathcal{O}_-(t) \equiv \left[\Psi_{t-\nu K,t-1}^TC_{t-\nu K}^T,\ \Psi_{t-\nu K+1,t-1}^TC_{t-\nu K+1}^T, \ldots,\ \Psi_{t-1,t-1}^TC_{t-1}^T\right]^T$$
$$(t = 0, 1, \ldots, K-1).$$

DEFINITION 4.2 (observability Gramians). [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

(i) [illegible] (1.1) [illegible]

$$G_k^{co} \equiv \mathcal{O}_+(k)^T\mathcal{O}_+(k) = \sum_{i=k}^{\infty} \Psi_{i,k-1}^T C_i^T C_i \Psi_{i,k-1} \in \mathbb{R}^{\mu_k \times \mu_k}, \quad k = 0, 1, \dots, K-1.$$

(ii) [illegible] (1.1) [illegible]

$$G_k^{no} \equiv \mathcal{O}_-(k)^T\mathcal{O}_-(k) = \sum_{i=k-\nu K}^{k-1} \Psi_{i,k-1}^T C_i^T C_i \Psi_{i,k-1} \in \mathbb{R}^{\mu_k \times \mu_k}, \quad k = 0, 1, \dots, K-1.$$

(iii) [illegible] (1.1) [illegible]

$$G_k^o \equiv G_k^{co} + G_k^{no}, \quad k = 0, 1, \dots, K-1.$$

[illegible] (i) From Definitions 4.1 and 4.2, the causal Gramians $G_k^{cr}$ and $G_k^{co}$ can be rewritten via (2.15) as

$$G_k^{cr} = Y_k \left( \sum_{i=-\infty}^{k-1} \Phi_f(k, i+1) B_i^f (B_i^f)^T \Phi_f(k, i+1)^T \right) Y_k^T,$$

$$G_k^{co} = X_k^T \left( \sum_{i=k}^{\infty} \Phi_f(i, k)^T (C_i^f)^T C_i^f \Phi_f(i, k) \right) X_k.$$

As mentioned in [45, 46], the infinite series in brackets above converge and are defined as reachability and observability Gramians for an asymptotically stable periodic system with $E_k = I_{n_{k+1}}$. Therefore, infinite series that appeared in the definition of Gramians $G_k^{cr}$ and $G_k^{co}$ converge because of the pd-stability of the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$.

(ii) The Gramians $G_k^{cr}$, $G_k^{nr}$, $G_k^{co}$, and $G_k^{no}$ are symmetric positive semidefinite matrices for all time instants $k$.

(iii) Definitions 4.1 and 4.2 are natural generalizations of the Gramians defined for descriptor systems with period $K = 1$; see, e.g., [1, 39].

(iv) In section 6, the reachability and observability Gramians $G_k^r$ and $G_k^o$ will be used to define the Hankel singular values, which are then utilized in section 7 for balancing transformations in balanced realization.

The following theorem indicates that these Gramians of the periodic descriptor system (1.1) satisfy some projected generalized discrete-time periodic Lyapunov equations with special right-hand sides.

THEOREM 4.1. [illegible] (1.1) [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

(i) [illegible] $\{G_k^{cr}\}_{k=0}^{K-1}$ [illegible] $\{G_k^{nr}\}_{k=0}^{K-1}$ [illegible]

$$\begin{aligned} E_k G_{k+1}^{cr} E_k^T - A_k G_k^{cr} A_k^T &= P_l(k) B_k B_k^T P_l(k)^T, \\ G_k^{cr} &= P_r(k) G_k^{cr} P_r(k)^T, \quad k = 0, 1, 2, \dots, K-1, \end{aligned} \tag{4.1}$$

[illegible]

$$E_kG^{nr}_{k+1}E^T_k - A_kG^{nr}_kA^T_k = -(I_{\mu_{k+1}} - P_l(k))B_kB^T_k(I_{\mu_{k+1}} - P_l(k))^T, \quad P_r(k)G^{nr}_k = 0, \quad k = 0, 1, 2, \dots, K-1, \tag{4.2}$$

[illegible] $G^{cr}_K \equiv G^{cr}_0$ [illegible] $G^{nr}_K \equiv G^{nr}_0$

(ii) [illegible] $\{G^{co}_k\}_{k=0}^{K-1}$ [illegible] $\{G^{no}_k\}_{k=0}^{K-1}$ [illegible]

$$E^T_{k-1}G^{co}_kE_{k-1} - A^T_kG^{co}_{k+1}A_k = P_r(k)^TC^T_kC_kP_r(k), \quad G^{co}_k = P_l(k-1)^TG^{co}_kP_l(k-1), \quad k = 0, 1, \dots, K-1, \tag{4.3}$$

[illegible]

$$E^T_{k-1}G^{no}_kE_{k-1} - A^T_kG^{no}_{k+1}A_k = -(I_{n_k} - P_r(k))^TC^T_kC_k(I_{n_k} - P_r(k)), \quad G^{no}_kP_l(k-1) = 0, \quad k = 0, 1, 2, \dots, K-1, \tag{4.4}$$

[illegible] $G^{co}_K \equiv G^{co}_0$ $G^{no}_K \equiv G^{no}_0$ $E_{-1} \equiv E_{K-1}$ [illegible] $P_l(-1) \equiv P_l(K-1)$

(iii) [illegible] $\{G^{r}_k\}_{k=0}^{K-1}$ [illegible] $\{G^{o}_k\}_{k=0}^{K-1}$ [illegible]

$$E_kG^{r}_{k+1}E^T_k - A_kG^{r}_kA^T_k = P_l(k)B_kB^T_kP_l(k)^T - (I_{\mu_{k+1}} - P_l(k))B_kB^T_k(I_{\mu_{k+1}} - P_l(k))^T, \quad P_r(k)G^{r}_k = G^{r}_kP_r(k)^T, \quad k = 0, 1, 2, \dots, K-1, \tag{4.5}$$

[illegible]

$$E^T_{k-1}G^{o}_kE_{k-1} - A^T_kG^{o}_{k+1}A_k = P_r(k)^TC^T_kC_kP_r(k) - (I_{n_k} - P_r(k))^TC^T_kC_k(I_{n_k} - P_r(k)), \quad P_l(k-1)^TG^{o}_k = G^{o}_kP_l(k-1), \quad k = 0, 1, 2, \dots, K-1, \tag{4.6}$$

[illegible] $G^{r}_K \equiv G^{r}_0$ $G^{o}_K \equiv G^{o}_0$ $E_{-1} \equiv E_{K-1}$ [illegible] $P_l(-1) \equiv P_l(K-1)$

[illegible] We shall verify only (4.1) here, and the other cases can be treated similarly. From (2.2) and (2.14), we can rewrite (4.1) into the following matrix equations:

$$G_{k+1,11} - A^f_kG_{k,11}(A^f_k)^T = B^f_k(B^f_k)^T, \tag{4.7}$$

$$G_{k+1,12}(E^b_k)^T - A^f_kG_{k,12} = 0, \tag{4.8}$$

$$E^b_kG_{k+1,21} - G_{k,21}(A^f_k)^T = 0, \tag{4.9}$$

$$E^b_kG_{k+1,22}(E^b_k)^T - G_{k,22} = 0, \tag{4.10}$$

where $Y_k^{-1}G^{cr}_kY_k^{-T} = \begin{bmatrix} G_{k,11} & G_{k,12} \\ G_{k,21} & G_{k,22} \end{bmatrix}$ with $G_{k,11} \in \mathbb{R}^{n^f_k \times n^f_k}$ and $G_{k,22} \in \mathbb{R}^{n^\infty_k \times n^\infty_k}$. Since $\{(E_k, A_k)\}$ are pd-stable, the matrices $J_k = A^f_{k+K-1}A^f_{k+K-2}\cdots A^f_k$ ($k = 0, 1, 2, \dots, K-1$) contain only eigenvalues lying inside the unit circle, and $N_k = E^b_kE^b_{k+1}\cdots E^b_{k+K-1}$ ($k = 0, 1, 2, \dots, K-1$) contain only zero eigenvalues. Therefore, (4.7) and (4.10) have unique symmetric solutions $G_{k,11}$ and $G_{k,22}$, respectively (see remark (ii) after Lemma 2.1). Equations (4.8) and (4.9) are solvable and have, for example, trivial solutions. It follows from $G^{cr}_k = P_r(k)G^{cr}_kP_r(k)^T$ that

$$G^{cr}_k = Y_k\begin{bmatrix} G_{k,11} & G_{k,12} \\ G_{k,21} & G_{k,22} \end{bmatrix}Y^T_k = Pr(k)G^{cr}_kP_r(k)^T = Y_k\begin{bmatrix} G_{k,11} & 0 \\ 0 & 0 \end{bmatrix}Y^T_k;$$

i.e., $G_{k,12} = G_{k,21} = G_{k,22} = 0$. Thus, the matrices

$$G_k^{cr} = Y_k \begin{bmatrix} G_{k,11} & 0 \\ 0 & 0 \end{bmatrix} Y_k^T$$

are the unique symmetric solutions of the projected GDPLEs (4.1) with $G_k^{cr} = P_r(k) G_k^{cr} P_r(k)^T$.

On the other hand, it can be shown that the causal reachability Gramians $G_k^{cr}$ satisfy the projected GDPLEs (4.1). Indeed, direct substitutions, using (2.1) and (2.15), give

$$\begin{aligned}
&E_k G_{k+1}^{cr} E_k^T - A_k G_k^{cr} A_k^T \\
&= E_k \left( \sum_{i=-\infty}^{k} \Psi_{k+1,i} B_i B_i^T \Psi_{k+1,i}^T \right) E_k^T - A_k \left( \sum_{i=-\infty}^{k-1} \Psi_{k,i} B_i B_i^T \Psi_{k,i}^T \right) A_k^T \\
&= E_k Y_{k+1} \left( \sum_{i=-\infty}^{k} \begin{bmatrix} \Phi_f(k+1,i+1) & 0 \\ 0 & 0 \end{bmatrix} X_i B_i B_i^T X_i^T \begin{bmatrix} \Phi_f(k+1,i+1)^T & 0 \\ 0 & 0 \end{bmatrix} \right) Y_{k+1}^T E_k^T \\
&\quad - A_k Y_k \left( \sum_{i=-\infty}^{k-1} \begin{bmatrix} \Phi_f(k,i+1) & 0 \\ 0 & 0 \end{bmatrix} X_i B_i B_i^T X_i^T \begin{bmatrix} \Phi_f(k,i+1)^T & 0 \\ 0 & 0 \end{bmatrix} \right) Y_k^T A_k^T \\
&= X_k^{-1} \begin{bmatrix} \sum_{-\infty}^{k} \Phi_f(k+1,i+1) B_i^f (B_i^f)^T \Phi_f(k+1,i+1)^T & 0 \\ 0 & 0 \end{bmatrix} X_k^{-T} \\
&\quad - X_k^{-1} \begin{bmatrix} \sum_{-\infty}^{k-1} \Phi_f(k+1,i+1) B_i^f (B_i^f)^T \Phi_f(k+1,i+1)^T & 0 \\ 0 & 0 \end{bmatrix} X_k^{-T} \\
&= X_k^{-1} \begin{bmatrix} B_k^f (B_k^f)^T & 0 \\ 0 & 0 \end{bmatrix} X_k^{-T} = P_l(k) B_k B_k^T P_l(k)^T
\end{aligned}$$

and

$$\begin{aligned}
&P_r(k) G_k^{cr} P_r(k)^T \\
&= Y_k \begin{bmatrix} I_{n_k^f} & 0 \\ 0 & 0 \end{bmatrix} Y_k^{-1} \left( \sum_{i=-\infty}^{k-1} \Psi_{k,i} B_i B_i^T \Psi_{k,i}^T \right) Y_k^T \begin{bmatrix} I_{n_k^f} & 0 \\ 0 & 0 \end{bmatrix} Y_k^T \\
&= Y_k \begin{bmatrix} \sum_{i=-\infty}^{k-1} \Phi_f(k,i+1) B_i^f (B_i^f)^T \Phi_f(k,i+1)^T & 0 \\ 0 & 0 \end{bmatrix} Y_k^T = G_k^{cr},
\end{aligned}$$

for $k = 0, 1, \dots, K-1$. Therefore, the causal reachability Gramians $\{G_k^{cr}\}_{k=0}^{K-1}$ are the unique symmetric positive semidefinite solutions of the projected GDPLEs (4.1). ☐

The following theorem shows that complete reachability/observability of the periodic descriptor system (1.1) can be characterized via the reachability/observability Gramians.

THEOREM 4.2. [illegible] (1.1) [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

(i) [illegible] (1.1) [illegible] $G_k^r$ [illegible] $k = 0, 1, 2, \dots, K-1$

(ii) [illegible] (1.1) [illegible] $G_k^o$ [illegible] $k = 0, 1, 2, \dots, K-1$

*Proof.* Here we shall prove only statement (i); statement (ii) can be verified similarly. For $k = 0, 1, \dots, K-1$, premultiply (4.5) by $X_k$ and postmultiply (4.5) by $X_k^T$; it follows that

(4.11)
$$X_k E_k Y_{k+1} \widehat{G}_{k+1}^r Y_{k+1}^T E_k^T X_k^T - X_k A_k Y_k \widehat{G}_k^r Y_k^T A_k^T X_k^T = \begin{bmatrix} B_k^f (B_k^f)^T & 0 \\ 0 & -B_k^b (B_k^b)^T \end{bmatrix},$$

where $\widehat{G}_k^r \equiv Y_k^{-1} G_k^r Y_k^{-T}$.

From Definition 4.1 it is easily seen, for $k = 0, 1, \dots, K-1$, that

$$\widehat{G}_k^r = Y_k^{-1} G_k^r Y_k^{-T} = \begin{bmatrix} \widehat{G}_{k,1}^{cr} & 0 \\ 0 & \widehat{G}_{k,2}^{nr} \end{bmatrix}, \tag{4.12}$$

with

$$\widehat{G}_{k,1}^{cr} \equiv \sum_{i=-\infty}^{k-1} \Phi_f(k, i+1) B_i^f (B_i^f)^T \Phi_f(k, i+1)^T,$$
$$\widehat{G}_{k,2}^{nr} \equiv \sum_{i=k}^{k+\nu K-1} \Phi_b(k, i) B_i^b (B_i^b)^T \Phi_b(k, i)^T\ .$$

Then by (2.1) and (4.12), (4.11) is decomposed into two periodic Lyapunov equations, for $k = 0, 1, 2, \dots, K-1$:

$$\widehat{G}_{k+1,1}^{cr} - A_k^f \widehat{G}_{k,1}^{cr} (A_k^f)^T = B_k^f (B_k^f)^T, \tag{4.13}$$
$$\widehat{G}_{k,2}^{nr} - E_k^b \widehat{G}_{k+1,2}^{nr} (E_k^b)^T = B_k^b (B_k^b)^T. \tag{4.14}$$

Rewrite (4.13) and (4.14) into two enlarged Lyapunov equations:

$$\mathcal{G}_{cr} - \mathcal{A}_f \mathcal{G}_{cr} \mathcal{A}_f^T = \mathcal{B}_f \mathcal{B}_f^T, \tag{4.15}$$
$$\mathcal{G}_{nr} - \mathcal{E}_b \mathcal{G}_{nr} \mathcal{E}_b^T = \mathcal{B}_b \mathcal{B}_b^T, \tag{4.16}$$

where $\mathcal{G}_{cr} = \operatorname{diag}(\widehat{G}_{k,1}^{cr}, \dots, \widehat{G}_{K-1,1}^{cr}, \widehat{G}_{0,1}^{cr})$, $\mathcal{G}_{nr} = \operatorname{diag}(\widehat{G}_{0,2}^{nr}, \widehat{G}_{1,2}^{nr}, \dots, \widehat{G}_{K-1,2}^{nr})$, $\mathcal{E}_b$, and $\mathcal{B}_b$ are as defined in (3.1), and

$$\mathcal{A}_f = \begin{bmatrix} & & & A_0^f \\ A_1^f & & & \\ & \ddots & & \\ & & A_{K-1}^f & \end{bmatrix}, \quad \mathcal{B}_f = \begin{bmatrix} B_0^f & & & \\ & B_1^f & & \\ & & \ddots & \\ & & & B_{K-1}^f \end{bmatrix}.$$

Since the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ are pd-stable and the matrix $\mathcal{E}_b$ is nilpotent with index $\nu$, the pairs $(\mathcal{A}_f, \mathcal{B}_f)$ and $(\mathcal{E}_b, \mathcal{B}_b)$ are reachable if and only if the solutions $\mathcal{G}_{cr}$ and $\mathcal{G}_{nr}$ of Lyapunov equations (4.15) and (4.16) are symmetric positive definite. Equivalently, following from (4.12), all reachability Gramians $G_k^r$ $(k = 0, 1, \dots, K-1)$ are symmetric positive definite. Moreover, from Theorems 3.1–3.2 and the remark following Definition 3.1, we know that the periodic descriptor system (1.1) is completely reachable if and only if the pairs $(\mathcal{A}_f, \mathcal{B}_f)$ and $(\mathcal{E}_b, \mathcal{B}_b)$ are reachable. This completes the proof of statement (i). ☐

**5. Numerical solutions of projected GDPLEs.** In this section, a numerical method is proposed for the symmetric positive semidefinite solutions of the projected generalized discrete-time periodic Lyapunov equations (4.1) and (4.3), for pd-stable $\{(E_k, A_k)\}_{k=0}^{K-1}$. We first consider the numerical solutions of the GDPLEs (4.3).

**GDPLEs for observability Gramians $\boldsymbol{G_k^{co}}$.** As $\{(E_k, A_k)\}_{k=0}^{K-1}$ are pd-stable, there exist orthogonal matrices $V_k \in \mathbb{R}^{\mu_{k+1}\times\mu_{k+1}}$ and $U_k \in \mathbb{R}^{n_k\times n_k}$, with $U_K \equiv U_0$, such that the decompositions (2.2) of system matrices $E_k$ and $A_k$ hold.

Notice that

$$\begin{bmatrix} I_{n_{k+1}^f} & Z_k \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} E_{k,1} & E_{k,3} \\ 0 & E_{k,2} \end{bmatrix} \begin{bmatrix} I_{n_{k+1}^f} & -W_{k+1} \\ 0 & I_{n_{k+1}^\infty} \end{bmatrix} = \begin{bmatrix} E_{k,1} & 0 \\ 0 & E_{k,2} \end{bmatrix}, \tag{5.1}$$

$$\begin{bmatrix} I_{n_{k+1}^f} & Z_k \\ 0 & I_{n_k^\infty} \end{bmatrix} \begin{bmatrix} A_{k,1} & A_{k,3} \\ 0 & A_{k,2} \end{bmatrix} \begin{bmatrix} I_{n_k^f} & -W_k \\ 0 & I_{n_k^\infty} \end{bmatrix} = \begin{bmatrix} A_{k,1} & 0 \\ 0 & A_{k,2} \end{bmatrix} \tag{5.2}$$

if the matrices $Z_k \in \mathbb{R}^{n_{k+1}^f\times n_k^\infty}$ and $W_k \in \mathbb{R}^{n_k^f\times n_k^\infty}$, with $W_K \equiv W_0$ and for $k = 0, 1, \ldots, K-1$, satisfy the generalized periodic Sylvester equations

$$\begin{aligned} E_{k,1}W_{k+1} - Z_kE_{k,2} &= E_{k,3}, \\ A_{k,1}W_k - Z_kA_{k,2} &= A_{k,3}. \end{aligned} \tag{5.3}$$

The generalized periodic Sylvester equations (5.3) have unique solutions $Z_k$ and $W_k$ (see Remark (ii) after Lemma 2.1). Therefore, the nonsingular matrices $X_k, Y_k$ in (2.1) satisfy

$$X_k = \begin{bmatrix} I_{n_{k+1}^f} & Z_k \\ 0 & I_{n_k^\infty} \end{bmatrix} V_k^T, \quad Y_k = U_k \begin{bmatrix} I_{n_k^f} & -W_k \\ 0 & I_{n_k^\infty} \end{bmatrix},$$

and the right and left spectral projections $P_r(k)$, $P_l(k)$ are given as

$$P_l(k) = V_k \begin{bmatrix} I_{n_{k+1}^f} & Z_k \\ 0 & 0 \end{bmatrix} V_k^T, \quad P_r(k) = U_k \begin{bmatrix} I_{n_k^f} & W_k \\ 0 & 0 \end{bmatrix} U_k^T. \tag{5.4}$$

Let, for $k = 0, 1, \ldots, K-1$,

$$V_{k-1}^T G_k^{co} V_{k-1} = \begin{bmatrix} G_{k,1}^{co} & G_{k,3}^{co} \\ (G_{k,3}^{co})^T & G_{k,2}^{co} \end{bmatrix}, \quad C_kU_k = \begin{bmatrix} C_{k,1}, & C_{k,2} \end{bmatrix}. \tag{5.5}$$

Substituting (2.2), (5.4), and (5.5) into the projected GDPLEs (4.3), for $k = 0, 1, \ldots, K-1$, we have

$$E_{k-1,1}^T G_{k,1}^{co} E_{k-1,1} - A_{k,1}^T G_{k+1,1}^{co} A_{k,1} = C_{k,1}^T C_{k,1}, \tag{5.6}$$

$$\begin{aligned} &E_{k-1,1}^T G_{k,1}^{co} E_{k-1,3} + E_{k-1,1}^T G_{k,3}^{co} E_{k-1,2} - A_{k,1}^T G_{k+1,1}^{co} A_{k,3} - A_{k,1}^T G_{k+1,3}^{co} A_{k,2} \\ &\quad = C_{k,1}^T C_{k,1} W_k, \end{aligned} \tag{5.7}$$

$$\begin{aligned} &E_{k-1,3}^T G_{k,1}^{co} E_{k-1,3} + E_{k-1,3}^T G_{k,3}^{co} E_{k-1,2} + E_{k-1,2}^T (G_{k,3}^{co})^T E_{k-1,3} \\ &\quad + E_{k-1,2}^T G_{k,2}^{co} E_{k-1,2} - A_{k,3}^T G_{k+1,1}^{co} A_{k,3} - A_{k,3}^T G_{k+1,3}^{co} A_{k,2} \\ &\quad - A_{k,2}^T (G_{k+1,3}^{co})^T A_{k,3} - A_{k,2}^T G_{k+1,2}^{co} A_{k,2} = W_k^T C_{k,1}^T C_{k,1} W_k. \end{aligned} \tag{5.8}$$

Again from the pd-stability of $\{(E_{k,1}, A_{k,1})\}_{k=0}^{K-1}$, the generalized discrete-time periodic Lyapunov equations (5.6) have unique symmetric positive semidefinite solutions $G_{k,1}^{co}$. Furthermore, it follows from (5.3) that (5.7) can be rearranged as

$$E_{k-1,1}^T(G_{k,3}^{co} - G_{k,1}^{co}Z_{k-1})E_{k-1,2} - A_{k,1}^T(G_{k+1,3}^{co} - G_{k+1,1}^{co}Z_k)A_{k,2} = 0. \tag{5.9}$$

We deduce that

$$G_{k,3}^{co} = G_{k,1}^{co}Z_{k-1}, \quad k = 0, 1, \ldots, K-1. \tag{5.10}$$

From (5.3), (5.6), and (5.10), now (5.8) can be rewritten as

$$E_{k-1,2}^T(G_{k,2}^{co} - Z_{k-1}^T G_{k,1}^{co}Z_{k-1})E_{k-1,2} - A_{k,2}^T(G_{k+1,2}^{co} - Z_k^T G_{k+1,1}^{co}Z_k)A_{k,2} = 0. \tag{5.11}$$

Since the periodic matrix pairs $\{(E_{k,2}, A_{k,2})\}_{k=0}^{K-1}$ have only infinite eigenvalues, we then have

$$G_{k,2}^{co} = Z_{k-1}^T G_{k,1}^{co}Z_{k-1}, \quad k = 0, 1, \ldots, K-1. \tag{5.12}$$

Therefore, the solutions of the projected GDPLEs (4.3) have the form

$$G_k^{co} = V_{k-1}\begin{bmatrix} G_{k,1}^{co} & G_{k,1}^{co}Z_{k-1} \\ Z_{k-1}^T G_{k,1}^{co} & Z_{k-1}^T G_{k,1}^{co}Z_{k-1} \end{bmatrix} V_{k-1}^T, \quad k = 0, 1, \ldots, K-1, \tag{5.13}$$

where the matrices $G_{k,1}^{co}$ are the unique symmetric positive semidefinite solutions of the generalized periodic Lyapunov equations (5.6) (see Remark (ii) after Lemma 2.1). Moreover, from (5.4) and (5.13), they also satisfy $P_l(k-1)^T G_k^{co} P_l(k-1) = G_k^{co}$.

In many applications it is necessary to have the Cholesky factors of the solutions of the Lyapunov equations rather the solutions themselves [24]. In particular, these full-rank factors are useful for numerically computing the Hankel singular values (see the next section). If $L_{k,1}$ denotes a Cholesky factor of each matrix $G_{k,1}^{co}$, i.e., $G_{k,1}^{co} = L_{k,1}^T L_{k,1}$, then we compute the QR factorization

$$L_{k,1} = Q_{k,L}\begin{bmatrix} T_{k,L} \\ 0 \end{bmatrix},$$

where $Q_{k,L}$ is orthogonal and $T_{k,L}$ has full row rank, for $k = 0, 1, \ldots, K-1$. The full-rank factorizations of the solutions $G_k^{co}$, for $k = 0, 1, \ldots, K-1$, are given by

$$\begin{aligned} G_k^{co} &= V_{k-1}\begin{bmatrix} L_{k,1}^T \\ Z_{k-1}^T L_{k,1}^T \end{bmatrix}\begin{bmatrix} L_{k,1}, & L_{k,1}Z_{k-1} \end{bmatrix} V_{k-1}^T \\ &= V_{k-1}\begin{bmatrix} T_{k,L}^T \\ Z_{k-1}^T T_{k,L}^T \end{bmatrix}\begin{bmatrix} T_{k,L}, & T_{k,L}Z_{k-1} \end{bmatrix} V_{k-1}^T \\ &\equiv L_k^T L_k, \end{aligned}$$

where $L_k \equiv \begin{bmatrix} T_{k,L}, & T_{k,L}Z_{k-1} \end{bmatrix} V_{k-1}^T$ has full row rank.

**GDPLEs for reachability Gramians $G_k^{cr}$.** Similarly for the projected GDPLEs (4.1), for $k = 0, 1, \ldots, K-1$, we let

$$U_k^T G_k^{cr} U_k = \begin{bmatrix} G_{k,1}^{cr} & G_{k,3}^{cr} \\ (G_{k,3}^{cr})^T & G_{k,2}^{cr} \end{bmatrix}, \quad V_k^T B_k = \begin{bmatrix} B_{k,1} \\ B_{k,2} \end{bmatrix}. \tag{5.14}$$

Substituting (2.2), (5.4), and (5.14) into the projected GDPLEs (4.1), we then have

$$\begin{aligned}E_{k,1}G^{cr}_{k+1,1}E^T_{k,1} - A_{k,1}G^{cr}_{k,1}A^T_{k,1} &= -E_{k,1}G^{cr}_{k+1,3}E^T_{k,3} - E_{k,3}(G^{cr}_{k+1,3})^T E^T_{k,1} - E_{k,3}G^{cr}_{k+1,2}E^T_{k,3}\\ &\quad + A_{k,1}G^{cr}_{k,3}A^T_{k,3} + A_{k,3}(G^{cr}_{k,3})^T A^T_{k,1} + A_{k,3}G^{cr}_{k,2}A^T_{k,3}\\ &\quad + (B_{k,1} + Z_k B_{k,2})(B_{k,1} + Z_k B_{k,2})^T,\end{aligned} \tag{5.15}$$

$$E_{k,1}G^{cr}_{k+1,3}E^T_{k,2} - A_{k,1}G^{cr}_{k,3}A^T_{k,2} = -E_{k,3}G^{cr}_{k+1,2}E^T_{k,2} + A_{k,3}G^{cr}_{k,2}A^T_{k,2}, \tag{5.16}$$

$$E_{k,2}G^{cr}_{k+1,2}E^T_{k,2} - A_{k,2}G^{cr}_{k,2}A^T_{k,2} = 0, \quad k = 0, 1, \dots, K-1. \tag{5.17}$$

Since the periodic matrix pairs $\{(E_{k,2}, A_{k,2})\}_{k=0}^{K-1}$ have only infinite eigenvalues, it follows from (5.17) that

$$G^{cr}_{k,2} = 0, \quad k = 0, 1, \dots, K-1. \tag{5.18}$$

Furthermore, (5.16) can be simplified to

$$E_{k,1}G^{cr}_{k+1,3}E^T_{k,2} - A_{k,1}G^{cr}_{k,3}A^T_{k,2} = 0. \tag{5.19}$$

We then have

$$G^{cr}_{k,3} = 0, \quad k = 0, 1, \dots, K-1. \tag{5.20}$$

From (5.18) and (5.20), (5.15) can be rewritten as

$$E_{k,1}G^{cr}_{k+1,1}E^T_{k,1} - A_{k,1}G^{cr}_{k,1}A^T_{k,1} = (B_{k,1} + Z_k B_{k,2})(B_{k,1} + Z_k B_{k,2})^T. \tag{5.21}$$

Therefore, the solutions of the projected GDPLEs (4.1) have the form

$$G^{cr}_k = U_k \begin{bmatrix} G^{cr}_{k,1} & 0 \\ 0 & 0 \end{bmatrix} U^T_k, \quad k = 0, 1, \dots, K-1, \tag{5.22}$$

where the matrices $G^{cr}_{k,1}$ are the unique symmetric positive semidefinite solutions of the generalized periodic Lyapunov equations (5.21) (see Remark (ii) after Lemma 2.1). Moreover, from (5.4) and (5.22), they also satisfy $P_r(k)G^{cr}_k P_r(k)^T = G^{cr}_k$.

If $R_{k,1}$ denotes a Cholesky factor of each matrix $G^{cr}_{k,1}$, i.e., $G^{cr}_{k,1} = R_{k,1}R^T_{k,1}$, then we compute the QR factorization

$$R^T_{k,1} = Q_{k,R} \begin{bmatrix} T^T_{k,R} \\ 0 \end{bmatrix},$$

where $Q_{k,R}$ is orthogonal and $T_{k,R}$ has full column rank. The full-rank factorizations of the solutions $G^{cr}_k$ are given by

$$\begin{aligned} G^{cr}_k &= U_k \begin{bmatrix} R_{k,1} \\ 0 \end{bmatrix} \begin{bmatrix} R^T_{k,1}, & 0 \end{bmatrix} U^T_k \\ &= U_k \begin{bmatrix} T_{k,R} \\ 0 \end{bmatrix} \begin{bmatrix} T^T_{k,R}, & 0 \end{bmatrix} U^T_k \\ &\equiv R_k R^T_k, \end{aligned}$$

where $R^T_k \equiv \begin{bmatrix} T^T_{k,R}, & 0 \end{bmatrix} U^T_k$ has full row rank for $k = 0, 1, \dots, K-1$.

**Algorithm GDPLE.** We now summarize the main steps for computing the full-rank Cholesky factors of the causal Gramians, via the solution of the GDPLEs (4.1) and (4.3). For simplicity in Algorithm 5.1, we shall ignore the obvious qualification for $k$, i.e., $k = 0, 1, \dots, K-1$.

ALGORITHM 5.1 (GDPLE).

**Input:** System matrices $(E_k, A_k, B_k, C_k)$, with $\{(E_k, A_k)\}_{k=0}^{K-1}$ being pd-stable.

**Output:** Full-rank Cholesky factors $R_k$ and $L_k$ ($k = 0, 1, \dots, K-1$), where $G_k^{cr} = R_k R_k^T$ and $G_k^{co} = L_k^T L_k$.

**Step 1.** Use the algorithm [48] to compute orthogonal matrices $V_k$ and $U_k$, with $U_K \equiv U_0$, such that

$$V_k^T E_k U_{k+1} = \begin{bmatrix} E_{k,1} & E_{k,3} \\ 0 & E_{k,2} \end{bmatrix}, \quad V_k^T A_k U_k = \begin{bmatrix} A_{k,1} & A_{k,3} \\ 0 & A_{k,2} \end{bmatrix},$$

where the matrices $E_{k,1}$ and $A_{k,2}$ are nonsingular, and

$$(A_{k,2})^{-1} E_{k,2} (A_{k+1,2})^{-1} E_{k+1,2} \cdots (A_{k+K-1,2})^{-1} E_{k+K-1,2}$$

are nilpotent.

**Step 2.** Compute the solutions of the generalized periodic Sylvester equations

$$\begin{aligned} E_{k,1} W_{k+1} - Z_k E_{k,2} &= E_{k,3}, \\ A_{k,1} W_k - Z_k A_{k,2} &= A_{k,3}. \end{aligned}$$

**Step 3.** Compute the matrices

$$V_k^T B_k = \begin{bmatrix} B_{k,1} \\ B_{k,2} \end{bmatrix}, \quad C_k U_k = \begin{bmatrix} C_{k,1}, & C_{k,2} \end{bmatrix}.$$

**Step 4.** Compute the Cholesky factors $R_{k,1}$ and $L_{k,1}$ of the solutions $G_{k,1}^{cr} = R_{k,1} R_{k,1}^T$ and $G_{k,1}^{co} = L_{k,1}^T L_{k,1}$ of the generalized discrete-time periodic Lyapunov equations

$$\begin{aligned} &E_{k,1} G_{k+1,1}^{cr} E_{k,1}^T - A_{k,1} G_{k,1}^{cr} A_{k,1}^T = (B_{k,1} + Z_k B_{k,2})(B_{k,1} + Z_k B_{k,2})^T, \\ &E_{k-1,1}^T G_{k,1}^{co} E_{k-1,1} - A_{k,1}^T G_{k+1,1}^{co} A_{k,1} = C_{k,1}^T C_{k,1}. \end{aligned}$$

**Step 5.** Compute the QR factorizations

$$R_{k,1}^T = Q_{k,R} \begin{bmatrix} T_{k,R}^T \\ 0 \end{bmatrix}, \quad L_{k,1} = Q_{k,L} \begin{bmatrix} T_{k,L} \\ 0 \end{bmatrix}.$$

**Step 6.** Compute the full-rank Cholesky factors

$$R_k = U_k \begin{bmatrix} T_{k,R} \\ 0 \end{bmatrix}, \quad L_k = \begin{bmatrix} T_{k,L}, & T_{k,L} Z_{k-1} \end{bmatrix} V_{k-1}^T.$$

[illegible]. One can extend the techniques in [30], for the numerical solution of the generalized Lyapunov equations, to solve the GDPLEs given in Step 4. A thorough error analysis and practical implementation details for the algorithm extended from [30] are still under investigation. It may also be possible to generalize the Hammarling-like method proposed in [36] to our periodic discrete-time case.

For Step 2, the periodic Sylvester equations can always be solved as an expanded linear equation using Kronecker products. Without such an expansion and for systems of constant dimensions, a more efficient algorithm has been proposed in [9]. For general systems of time-varying dimensions, the proof of Lemma 2.1 provides the theoretical basis for the algorithm. The heart of the algorithm lies in the computation of the Kronecker-like canonical form in [48]. The analysis and implementation of such an algorithm will be an interesting future project.

**A numerical example.** We shall illustrate the feasibility and reliability of the proposed algorithm with an example. All computations were performed in MATLAB/ version 6.5 on a PC with an Intel Pentium-III processor at 866 MHz, with 768 MB RAM, using IEEE double-precision floating-point arithmetic. The machine precision is approximately $2.22 \times 10^{-16}$. In our example, $E_k$ and $A_k$ are $n \times n$ square matrices, where $\mu_k = n_k = n$ for all $k$. Thus, instead of using the algorithm in [48] for periodic systems of time-varying dimensions, we utilize the PQZ algorithm with eigenvalue reordering strategy [7, 19] in Step 1 of the proposed algorithm GDPLE. The other steps are the same as those for periodic systems of constant dimensions.

For approximate solutions $\widetilde{X}_k$ of the projected GDPLEs (4.1) and (4.3), we compute the relative residuals defined by

$$\gamma_k^{cr} = \frac{\|E_k \widetilde{X}_{k+1} E_k^T - A_k \widetilde{X}_k A_k^T - P_l(k) B_k B_k^T P_l(k)^T\|_2}{\|\widetilde{X}_k\|_2},$$

$$\gamma_k^{co} = \frac{\|E_{k-1}^T \widetilde{X}_k E_{k-1} - A_k^T \widetilde{X}_{k+1} A_k - P_r(k)^T C_k^T C_k P_r(k)\|_2}{\|\widetilde{X}_k\|_2}.$$

[illegible] 1. We consider a periodic discrete-time descriptor system (1.1) with $n = 10$, $m = 2$, $p = 3$ and period $K = 3$. For $k = 0, 1, 2$, we have

$$E_k^{(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & c_1 & s_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -s_1 & c_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_1 & s_1 & 1 & 0 & c_2 & s_2 & 0 & 0 & 0 \\ 0 & -s_1 & c_1 & 0 & 1 & -s_2 & c_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_2 & s_2 & 1 & 0 & c_3 & s_3 & 0 \\ 0 & 0 & 0 & -s_2 & c_2 & 0 & 1 & -s_3 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_3 & s_3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -s_3 & c_3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$A_k^{(0)} = \operatorname{diag}(1.01, A_{01}, A_{02}, A_{03}, A_{04}, 1.001), \quad \theta_k := 2\pi k/K,$$

$$B_k^T = \begin{bmatrix} 4 & -1 & 3 & 5 & 0 & -2 & 0 & 8 & 1 & 0 \\ 1 & 1 & s_1+1 & -2 & 1 & 0 & 0 & -3 & 0 & 1 \end{bmatrix},$$

$$C_k = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.05+c_1 & 0 & 0 \end{bmatrix},$$

where

$$
\begin{aligned}
c_1 &= \cos(\theta_k), \quad c_2 = 0.2c_1, \quad c_3 = 0.6c_1,\\
s_1 &= \sin(\theta_k), \quad s_2 = 0.2s_1, \quad s_3 = 0.6s_1,
\end{aligned}
$$

$$
A_{01} = \begin{bmatrix} r_1\cos(\pi/3) & r_1\sin(\pi/3) \\ -r_1\sin(\pi/3) & r_1\cos(\pi/3) \end{bmatrix}, \quad A_{02} = \begin{bmatrix} r_2\cos(7\pi/5) & r_2\sin(7\pi/5) \\ -r_2\sin(7\pi/5) & r_2\cos(7\pi/5) \end{bmatrix},
$$

$$
A_{03} = \begin{bmatrix} r_3\cos(\pi/4) & r_3\sin(\pi/4) \\ -r_3\sin(\pi/4) & r_3\cos(\pi/4) \end{bmatrix}, \quad A_{04} = \begin{bmatrix} r_4\cos(\pi/10) & r_4\sin(\pi/10) \\ -r_4\sin(\pi/10) & r_4\cos(\pi/10) \end{bmatrix},
$$

and

$$
r_1 = 0.5, \quad r_2 = 0.05, \quad r_3 = -0.02, \quad r_4 = 0.12.
$$

We define a Householder transformation $V = I - 2uu^T$ with $u = [1, 1, \ldots, 1, 1]^T/\sqrt{10} \in \mathbb{R}^{10}$, and the $K$-periodic system matrices $(E_k, A_k, B_k, C_k)$ are given by

$$
E_k \equiv V^T E_k^{(0)} V, \quad A_k \equiv V^T A_k^{(0)} V, \quad k = 0, 1, 2.
$$

The computed open-loop spectrum of the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ consists of two infinite eigenvalues and four pairs of complex conjugate finite eigenvalues lying inside the unit circle. Thus, the periodic matrix pairs $\{(E_k, A_k)\}_{k=0}^{K-1}$ are pd-stable where $n_k^f = 8$ and $n_k^\infty = 2$ for all $k$. Accurate numerical results, indicated by small relative residuals, were produced by the proposed algorithm, as shown in Table 5.1.

TABLE 5.1
*Norms and relative residuals of causal Gramians.*

| $k$ | $\lVert G_k^{cr} \rVert_2$ | $\gamma_k^{cr}$ | $\lVert G_k^{co} \rVert_2$ | $\gamma_k^{co}$ |
|---|---|---|---|---|
| 0 | $8.30 \times 10^4$ | $2.17 \times 10^{-16}$ | $1.14 \times 10^3$ | $1.39 \times 10^{-16}$ |
| 1 | $7.11 \times 10^3$ | $3.11 \times 10^{-16}$ | $9.70 \times 10^0$ | $4.17 \times 10^{-15}$ |
| 2 | $5.82 \times 10^2$ | $6.73 \times 10^{-16}$ | $9.74 \times 10^1$ | $9.18 \times 10^{-15}$ |

## 6. Hankel singular values.

Similar to standard state space systems [17] and continuous-time descriptor systems [34, 38], the controllability and observability Gramians can be used to define Hankel singular values for the periodic descriptor systems (1.1), which are of great importance in the model reduction problem via the balanced truncation method.

For the discrete-time descriptor systems, the causal and noncausal Hankel singular values are defined via the nonnegative eigenvalues of the matrices $\mathcal{G}_{dcc}E^T\mathcal{G}_{dco}E$ and $\mathcal{G}_{dnc}A^T\mathcal{G}_{dno}A$. Here $\mathcal{G}_{dcc}$, $\mathcal{G}_{dnc}$, $\mathcal{G}_{dco}$, and $\mathcal{G}_{dno}$ denote the causal/noncausal reachability Gramians and the causal/noncausal observability Gramians, respectively [39].

LEMMA 6.1. [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] $n_k \times n_k$ [illegible] $\mathbf{H}_k^c \equiv G_k^{cr}E_{k-1}^T G_k^{co} E_{k-1}$ [illegible] $\mathbf{H}_k^{nc} \equiv G_k^{nr} A_k^T G_{k+1}^{no} A_k$ $k = 0, 1, 2, \ldots, K-1$ [illegible]

[illegible] From Definitions 4.1 and 4.2 and (2.15), for $k = 0, 1, 2, \ldots, K-1$, we have

$$
\mathbf{H}_k^c = Y_k \begin{bmatrix} \widehat{G}_{k,1}^{cr}\widehat{G}_{k,1}^{co} & 0 \\ 0 & 0 \end{bmatrix} Y_k^{-1},
$$

where

$$\widehat{G}_{k,1}^{cr} \equiv \sum_{i=-\infty}^{k-1} \Phi_f(k,i+1)B_i^f(B_i^f)^T\Phi_f(k,i+1)^T,$$

$$\widehat{G}_{k,1}^{co} \equiv \sum_{i=k}^{\infty} \Phi_f(i,k)^T(C_i^f)^TC_i^f\Phi_f(i,k).$$

Since the $n_k^f \times n_k^f$ matrices $\widehat{G}_{k,1}^{cr}$ and $\widehat{G}_{k,1}^{co}$ are symmetric positive semidefinite, it follows that all $\mathbf{H}_k^c$ have real and nonnegative eigenvalues. Similarly, it can be shown that all $\mathbf{H}_k^{nc}$ also share the same property. □

Notice that, in the proof of Lemma 6.1, $\mathbf{H}_k^c$ and $\mathbf{H}_k^{nc}$ have at least $n_k^\infty$ and $n_k^f$ zero eigenvalues, respectively. Hence, we have the following definition of Hankel singular values for the periodic descriptor system (1.1).

Definition 6.1. [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] $n_k^f$ $n_k^\infty$ [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

(i) [illegible] $k = 0, 1, \ldots, K-1$ [illegible] $n_k^f$ [illegible] $\mathbf{H}_k^c$ [illegible] $\zeta_{k,j}$ [illegible] (1.1)

(ii) [illegible] $k = 0, 1, \ldots, K-1$ [illegible] $n_k^\infty$ [illegible] $\mathbf{H}_k^{nc}$ [illegible] $\theta_{k,j}$ [illegible] (1.1)

[illegible] (i) When $K = 1$, the causal and noncausal Hankel singular values defined in Definition 6.1 coincide with those for discrete-time descriptor systems (see [39] and references therein). For $E_k = I$, the causal Hankel singular values are the classical Hankel singular values of linear periodic discrete-time systems [46].

(ii) As in the case of descriptor systems, the causal and noncausal Hankel singular values of the periodic descriptor system (1.1) are invariant under system equivalence transformations.

From Theorem 4.2 and Lemma 6.1, we obtain the following result.

Corollary 6.2. [illegible] (1.1) [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible]

(a) [illegible] (1.1) [illegible]

(b) [illegible] $k = 0, 1, 2, \ldots, K-1$ [illegible]

$$\operatorname{rank}(G_k^{cr}) = \operatorname{rank}(G_k^{co}) = \operatorname{rank}(\mathbf{H}_k^c) = n_k^f,$$
$$\operatorname{rank}(G_k^{nr}) = \operatorname{rank}(G_k^{no}) = \operatorname{rank}(\mathbf{H}_k^{nc}) = n_k^\infty.$$

(c) [illegible] (1.1) [illegible]

For pd-stable $\{(E_k, A_k)\}_{k=0}^{K-1}$, the causal and noncausal reachability and observability Gramians are symmetric and positive semidefinite. Thus, there exist full-rank factorizations

$$\begin{aligned} G_k^{cr} &= R_kR_k^T, \quad G_k^{co} = L_k^TL_k, \\ G_k^{nr} &= \widetilde{R}_k\widetilde{R}_k^T, \quad G_k^{co} = \widetilde{L}_k^T\widetilde{L}_k, \end{aligned} \tag{6.1}$$

where the matrices $R_k$, $L_k^T$, $\tilde{R}_k$, and $\tilde{L}_k^T$ are of full column rank.

The connections between the causal/noncausal Hankel singular values and the singular values of the matrices $L_kE_{k-1}R_k$ and $\widetilde{L}_{k+1}A_k\widetilde{R}_k$ are considered in the following lemma.

LEMMA 6.3. [illegible] (1.1) [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] (1.1) [illegible] (6.1) [illegible] $k = 0, 1, 2, \ldots, K-1$ [illegible] $L_kE_{k-1}R_k$ [illegible] $\widetilde{L}_{k+1}A_k\widetilde{R}_k$[illegible]

[illegible] Notice that for $k = 0, 1, \ldots, K-1$, we have

$$\zeta_{k,j}^2 = \lambda_j(R_kR_k^TE_{k-1}^TL_k^TL_kE_{k-1}) = \lambda_j(R_k^TE_{k-1}^TL_k^TL_kE_{k-1}R_k) = \sigma_j^2(L_kE_{k-1}R_k),$$

$$\theta_{k,j}^2 = \lambda_j(\widetilde{R}_k\widetilde{R}_k^TA_k^T\widetilde{L}_{k+1}^T\widetilde{L}_{k+1}A_k) = \lambda_j(\widetilde{R}_k^TA_k^T\widetilde{L}_{k+1}^T\widetilde{L}_{k+1}A_k\widetilde{R}_k) = \sigma_j^2(\widetilde{L}_{k+1}A_k\widetilde{R}_k),$$

where $\lambda_j(\cdot)$ and $\sigma_j(\cdot)$ denote respectively the eigenvalues and singular values of the corresponding matrices. ☐

## 7. Balanced realization.

It is well known [17] that for any minimal realization $(A, B, C)$ of a stable continuous-time or discrete-time system there exists a transformation such that the controllability and observability Gramians for the transformed realization equal some diagonal matrix. Such a realization is called a(n) (internally) balanced realization. Recently, the issues of balanced realization and model reduction via the balanced truncation method have been discussed for continuous-time descriptor systems [34, 38] and asymptotically stable linear discrete-time periodic systems [45, 46]. In this section the problem of balanced realization is generalized for periodic descriptor systems. We shall assume that the periodic descriptor system (1.1) is completely reachable/observable, with $\{(E_k, A_k)\}_{k=0}^{K-1}$ being pd-stable.

DEFINITION 7.1. [illegible] $(E_k, A_k, B_k, C_k)$ [illegible] (1.1) [illegible]

$$G_k^{cr} = G_k^{co} = \begin{bmatrix} D_{k,1} & 0 \\ 0 & 0 \end{bmatrix} \quad [illegible] \quad G_k^{nr} = G_{k+1}^{no} = \begin{bmatrix} 0 & 0 \\ 0 & D_{k,2} \end{bmatrix},$$

[illegible] $D_{k,1}$ [illegible] $D_{k,2}$ [illegible] $k = 0, 1, \ldots, K-1$

We shall show that for a realization $(E_k, A_k, B_k, C_k)$ of the periodic descriptor system (1.1) there exist nonsingular periodic matrices $S_k \in \mathbb{R}^{\mu_{k+1}\times\mu_{k+1}}$ and $T_k \in \mathbb{R}^{n_k\times n_k}$ $(k = 0, 1, \ldots, K-1)$ with $T_K \equiv T_0$, such that the transformed realization

$$(\widehat{E}_k, \widehat{A}_k, \widehat{B}_k, \widehat{C}_k) \equiv (S_k^TE_kT_{k+1}, S_k^TA_kT_k, S_k^TB_k, C_kT_k) \tag{7.1}$$

is balanced.

Consider the full-rank factorizations (6.1) of the causal/noncausal reachability/observability Gramians. For $k = 0, 1, \ldots, K-1$, let

$$L_kE_{k-1}R_k = U_k\Sigma_kV_k^T, \quad \widetilde{L}_{k+1}A_k\widetilde{R}_k = \widetilde{U}_k\Theta_k\widetilde{V}_k^T \tag{7.2}$$

be singular value decompositions [18], where $U_k$, $V_k$, $\widetilde{U}_k$, $\widetilde{V}_k$ are orthogonal and $\Sigma_k$ and $\Theta_k$ are diagonal and nonsingular. From Corollary 6.2 and Lemma 6.3, we have $\Sigma_k = \mathrm{diag}(\zeta_{k,1}, \ldots, \zeta_{k,n_k^f}) > 0$ and $\Theta_k = \mathrm{diag}(\theta_{k,1}, \ldots, \theta_{k,n_k^\infty}) > 0$. Furthermore, it is easily seen from Theorem 4.1 and (2.14) that

$$\begin{aligned}&G_k^{cr} = P_r(k) G_k^{cr} P_r(k)^T, \quad G_k^{co} = P_l(k-1)^T G_k^{co} P_l(k-1),\\ &P_r(k) G_k^{nr} = 0, \quad G_k^{no} P_l(k-1) = 0,\\ &E_{k-1} P_r(k) = P_l(k-1) E_{k-1}, \quad A_k P_r(k) = P_l(k) A_k.\end{aligned}$$

Simple calculations then yield $G_k^{no} E_{k-1} G_k^{cr} = G_k^{co} E_{k-1} G_k^{nr} = G_{k+1}^{no} A_k G_k^{cr} = G_{k+1}^{co} A_k G_k^{nr} = 0$. Hence, for $k = 0, 1, \ldots, K-1$, we have

$$\widetilde{L}_k E_{k-1} R_k = L_k E_{k-1} \widetilde{R}_k = \widetilde{L}_{k+1} A_k R_k = L_{k+1} A_k \widetilde{R}_k = 0. \tag{7.3}$$

Now for $k = 0, 1, \ldots, K-1$, consider the $\mu_{k+1} \times \mu_{k+1}$ matrices

$$\begin{aligned}S_k &= \begin{bmatrix} L_{k+1}^T U_{k+1} \Sigma_{k+1}^{-1/2}, & \widetilde{L}_{k+1}^T \widetilde{U}_k \Theta_k^{-1/2} \end{bmatrix},\\ \check{S}_k &= \begin{bmatrix} E_k R_{k+1} V_{k+1} \Sigma_{k+1}^{-1/2}, & A_k \widetilde{R}_k \widetilde{V}_k \Theta_k^{-1/2} \end{bmatrix};\end{aligned}$$

it follows from (7.2) and (7.3) that

$$\begin{aligned}S_k^T \check{S}_k &= \begin{bmatrix} \Sigma_{k+1}^{-1/2} U_{k+1}^T L_{k+1} E_k R_{k+1} V_{k+1} \Sigma_{k+1}^{-1/2} & \Sigma_{k+1}^{-1/2} U_{k+1}^T L_{k+1} A_k \widetilde{R}_k \widetilde{V}_k \Theta_k^{-1/2} \\ \Theta_k^{-1/2} \widetilde{U}_k^T \widetilde{L}_{k+1} E_k R_{k+1} V_{k+1} \Sigma_{k+1}^{-1/2} & \Theta_k^{-1/2} \widetilde{U}_k^T \widetilde{L}_{k+1} A_k \widetilde{R}_k \widetilde{V}_k \Theta_k^{-1/2} \end{bmatrix}\\ &= I_{\mu_{k+1}},\end{aligned}$$

i.e., the matrices $S_k$ and $\check{S}_k$ are nonsingular and $S_k^{-1} = \check{S}_k^T$. Similarly, it can be shown that the $n_k \times n_k$ matrices

$$T_k = \begin{bmatrix} R_k V_k \Sigma_k^{-1/2}, & \widetilde{R}_k \widetilde{V}_k \Theta_k^{-1/2} \end{bmatrix}, \quad \check{T}_k = \begin{bmatrix} E_{k-1}^T L_k^T U_k \Sigma_k^{-1/2}, & A_k^T \widetilde{L}_{k+1}^T \widetilde{U}_k \Theta_k^{-1/2} \end{bmatrix}$$

are also nonsingular and $T_k^{-1} = \check{T}_k^T$. Therefore, with the transformation matrices $S_k$ and $T_k$ defined above and (7.3), the causal reachability and observability Gramians of the transformed periodic descriptor system (7.1) become

$$\begin{aligned}\widehat{G}_k^{cr} &\equiv T_k^{-1} G_k^{cr} T_k^{-T} = \check{T}_k^T G_k^{cr} \check{T}_k\\ &= \begin{bmatrix} \Sigma_k^{-1/2} U_k^T L_k E_{k-1} R_k R_k^T E_{k-1}^T L_k^T U_k \Sigma_k^{-1/2} & \Sigma_k^{-1/2} U_k^T L_k E_{k-1} R_k R_k^T A_k^T \widetilde{L}_{k+1}^T \widetilde{U}_k \Theta_k^{-1/2} \\ \Theta_k^{-1/2} \widetilde{U}_k^T \widetilde{L}_{k+1} A_k R_k R_k^T E_{k-1}^T L_k^T U_k \Sigma_k^{-1/2} & \Theta_k^{-1/2} \widetilde{U}_k^T \widetilde{L}_{k+1} A_k R_k R_k^T A_k^T \widetilde{L}_{k+1}^T \widetilde{U}_k \Theta_k^{-1/2} \end{bmatrix}\\ &= \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix}\end{aligned}$$

and

$$\widehat{G}_k^{co} \equiv S_{k-1}^{-1} G_k^{co} S_{k-1}^{-T} = \check{S}_{k-1}^T G_k^{cr} \check{S}_{k-1}$$

$$= \begin{bmatrix} \Sigma_k^{-1/2} V_k^T R_k^T E_{k-1}^T L_k^T L_k E_{k-1} R_k V_k \Sigma_k^{-1/2} & \Sigma_k^{-1/2} V_k^T R_k^T E_{k-1}^T L_k^T L_k A_{k-1} \widetilde{R}_{k-1} \widetilde{V}_{k-1} \Theta_{k-1}^{-1/2} \\ \Theta_{k-1}^{-1/2} \widetilde{V}_{k-1}^T \widetilde{R}_{k-1}^T A_{k-1}^T L_k^T L_k E_{k-1} R_k V_k \Sigma_k^{-1/2} & \Theta_{k-1}^{-1/2} \widetilde{V}_{k-1}^T \widetilde{R}_{k-1}^T A_{k-1}^T L_k^T L_k A_{k-1} \widetilde{R}_{k-1} \widetilde{V}_{k-1} \Theta_{k-1}^{-1/2} \end{bmatrix}$$

$$= \begin{bmatrix} \Sigma_k & 0 \\ 0 & 0 \end{bmatrix}.$$

On the other hand, one can also show that the noncausal reachability and observability Gramians of the transformed periodic descriptor system (7.1) satisfy

$$\widehat{G}_k^{nr} \equiv T_k^{-1} G_k^{nr} T_k^{-T} = \left[ \begin{array}{cc} 0 & 0 \\ 0 & \Theta_k \end{array} \right] = S_k^{-1} G_{k+1}^{no} S_k^{-T} \equiv \widehat{G}_{k+1}^{no}, \quad k = 0, 1, \ldots, K-1.$$

Consequently, $S_k$ and $T_k$ ($k = 0, 1, \ldots, K-1$) are the desired balancing transformations such that the realization (7.1) is balanced. In summary, we have the following theorem.

THEOREM 7.1. [illegible] (1.1) [illegible] $\{(E_k, A_k)\}_{k=0}^{K-1}$ [illegible] $S_k$ [illegible] $T_k$ $k = 0, 1, \ldots, K-1$ [illegible] $T_K \equiv T_0$ [illegible] (7.1) [illegible]

[illegible] As in the cases of standard state space systems [17, 29] and descriptor systems [34, 38], the balancing transformation matrices for periodic descriptor system (1.1) are not unique. Indeed, if $\{(S_k, T_k)\}_{k=0}^{K-1}$ denotes a set of balancing transformation pairs for the periodic descriptor system (1.1), then for any diagonal matrix $D$ with diagonal entries $\pm 1$, the set of matrix pairs $\{(S_k D, T_k D)\}_{k=0}^{K-1}$ are also balancing transformation matrices for the periodic descriptor system (1.1).

## 8. Concluding remarks.

In Theorem 4.1, the reachability/observability Gramians are shown to satisfy some projected GDPLEs and can be computed numerically by applying Algorithm 5.1. We have developed the concepts of reachability/observability Gramians, Hankel singular values, and balanced realization for periodic discrete-time descriptor systems, based on the necessary and sufficient conditions for complete reachability and observability. These are useful in the model reduction problem via the balanced truncation method. A numerical example was given in section 5 to illustrate the feasibility and reliability of the proposed algorithm.

The theoretical development in the paper deals with general time-varying periodic systems. Understandably, numerical algorithms are more developed for systems of constant dimensions. Development of numerical algorithms (e.g., the solution of periodic Sylvester and Lyapunov equations) needs to be investigated further.

**Acknowledgment.** We would like to thank the referees for their penetrating questions, which improved the quality of the paper tremendously.

## REFERENCES

[1] D. J. BENDER, *Lyapunov-like equations and reachability/observability Gramians for descriptor systems*, IEEE Trans. Automat. Control, 32 (1987), pp. 343–348.

[2] M. C. BERG, N. AMIT, AND J. D. POWELL, *Multirate digital control system design*, IEEE Trans. Automat. Control, 33 (1988), pp. 1139–1150.

[3] S. BITTANTI, *Deterministic and stochastic linear periodic systems*, in Time Series and Linear Systems, S. Bittanti, ed., Springer-Verlag, New York, 1986, pp. 141–182.

[4] S. BITTANTI AND P. COLANERI, *Analysis of discrete-time linear periodic systems*, in Control and Dynamic Systems, Vol. 78, C. T. Leondes, ed., Academic Press, New York, 1996, pp. 313–339.

[5] S. BITTANTI AND P. COLANERI, *Periodic control*, in Wiley Encyclopedia of Electrical and Electronic Engineering, Vol. 16, J. G. Webster, ed., Wiley, New York, 1999, pp. 59–74.

[6] S. BITTANTI, P. COLANERI, AND G. D. NICOLAO, *The difference periodic Riccati equation for the periodic prediction problem*, IEEE Trans. Automat. Control, 33 (1988), pp. 706–712.

[7] A. BOJANCZYK, G. H. GOLUB, AND P. VAN DOOREN, *The periodic Schur decomposition. Algorithms and applications*, in Proc. SPIE, 1770 (1992), pp. 31–42.

[8] R. BRU, C. COLL, AND N. THOME, *Compensating periodic descriptor systems*, Systems Control Lett., 43 (2001), pp. 133–139.

[9] R. Byers and N. Rhee, *Cyclic Schur and Hessenberg Schur Numerical Methods for Solving Periodic Lyapunov and Sylvester Equations*, technical report, Department of Mathematics, University of Missouri at Kansas City, Kansas City, MO, 1995.

[10] E. K.-W. Chu, H.-Y. Fan, and W.-W. Lin, *Reachability and Observability of Periodic Descriptor Systems*, Preprint 2005-1-009, NCTS, National Tsing Hua University, Hsinchu 300, Taiwan, 2005.

[11] R. E. Crochiere and L. R. Rabiner, *Mutirate Digital Signal Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[12] L. Dai, *Singular Control Systems*, Springer-Verlag, Berlin, Heidelberg, 1989.

[13] A. Feuer and G. C. Goodwin, *Sampling in Digital Signal Processing and Control*, Birkhäuser, Boston, 1996.

[14] D. S. Flamm, *A new shift-invariant representation of periodic linear systems*, Systems Control Lett., 17 (1991), pp. 9–14.

[15] B. Francis and T. T. Georgiou, *Stability theory for linear time-invariant plants with periodic digital controllers*, IEEE Trans. Automat. Control, 33 (1988), pp. 820–832.

[16] W. A. Gardner, ed., *Cyclostationarity in Communications and Signal Processing*, IEEE Press, New York, 1994.

[17] K. Glover, *All optimal Hankel norm approximations of linear multivariable systems and their $L^\infty$ error bounds*, Internat. J. Control, 39 (1984), pp. 1115–1193.

[18] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.

[19] J. J. Hench and A. J. Laub, *Numerical solution of the discrete-time periodic Riccati equation*, IEEE Trans. Automat. Control, 39 (1994), pp. 1197–1210.

[20] R. W. Isniewski and M. Blanke, *Fully magnetic attitude control for spacecraft subject to gravity gradient*, Automatica, 35 (1999), pp. 1201–1214.

[21] W. Johnson, *Helicopter Theory*, Princeton University Press, Princeton, NJ, 1996.

[22] M. Kono, *Eigenvalue assignment in linear discrete-time system*, Internat. J. Control, 32 (1980), pp. 149–158.

[23] Y.-C. Kuo, W.-W. Lin, and S.-F. Xu, *Regularization of singular linear discrete-time periodic descriptor systems by derivative and proportional state feedback*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1046–1073.

[24] A. J. Laub, M. T. Heath, C. C. Paige, and R. C. Ward, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Trans. Automat. Control, 32 (1987), pp. 115–122.

[25] F. L. Lewis, *Fundamental, reachability, and observability matrices for discrete descriptor systems*, IEEE Trans. Automat. Control, 30 (1985), pp. 502–505.

[26] M.-L. Liou and Y.-L. Kuo, *Exact analysis of switched capacitor circuits with arbitrary inputs*, IEEE Trans. Circuits Systems, 26 (1979), pp. 213–223.

[27] A. Marzollo, *Periodic Optimization*, Springer-Verlag, Berlin, 1972.

[28] R. McKillip, *Periodic model following controller for the control-configured helicopter*, J. Amer. Helicopter Soc., 36 (1991), pp. 4–12.

[29] B. C. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–32.

[30] T. Penzl, *Numerical solution of generalized Lyapunov equations*, Adv. Comput. Math., 8 (1998), pp. 33–48.

[31] J. A. Richards, *Analysis of Periodically Time-Varying Systems*, Springer-Verlag, Berlin, 1983.

[32] V. V. Sergeichuk, *Computation of canonical matrices for chains and cycles of linear mappings*, Linear Algebra Appl., 376 (2004), pp. 235–263.

[33] J. Sreedhar and P. Van Dooren, *Periodic descriptor systems: Solvability and conditionability*, IEEE Trans. Automat. Control, 44 (1999), pp. 310–313.

[34] T. Stykel, *Model Reduction of Descriptor Systems*, Technical Report 720-2001, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, 2001.

[35] T. Stykel, *Analysis and Numerical Solution of Generalized Lyapunov Equations*, Ph.D. thesis, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, 2002.

[36] T. Stykel, *Numerical solution and perturbation theory for generalized Lyapunov equations*, Linear Algebra Appl., 349 (2002), pp. 155–185.

[37] T. Stykel, *Stability and inertia theorems for generalized Lyapunov equations*, Linear Algebra Appl., 355 (2002), pp. 297–314.

[38] T. Stykel, *Balanced Truncation Model Reduction for Semidiscretized Stokes Equation*, Technical Report 04-2003, Institut für Mathematik, Technische Universität Berlin, Berlin, Germany, 2003.

[39] T. Stykel, *Input-Output Invariants for Descriptor Systems*, Preprint PIMS-03-1, Pacific Institute for the Mathematical Sciences, Vancouver, BC, Canada, 2003.

[40] L. Tong, G. Xu, and T. Kailath, *Blind identification and equalization based on second-order statistics: A time domain approach*, IEEE Trans. Inform. Theory, 40 (1994), pp. 340–349.

[41] P. P. Vaidyanathan, *Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial*, Proc. IEEE, 78 (1990), pp. 56–93.

[42] P. P. Vaidyanathan, *Mutirate Systems and Filter-Banks*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[43] P. Van Dooren and J. Sreedhar, *When is a periodic discrete-time system equivalent to a time invariant one?*, Linear Algebra Appl., 212/213 (1994), pp. 131–151.

[44] A. Varga, *Periodic Lyapunov equations: Some applications and new algorithms*, Internat. J. Control, 67 (1997), pp. 69–87.

[45] A. Varga, *Balancing related methods for minimal realization of periodic systems*, Systems Control Lett., 36 (1999), pp. 339–349.

[46] A. Varga, *Balanced truncation model reduction of periodic systems*, in Proceedings of the IEEE Conference on Decision and Control, Sydney, Australia, 2000, IEEE Press, Piscataway, NJ, 2000, pp. 2379–2384.

[47] A. Varga, *Robust and minimum norm pole assignment with periodic state feedback*, IEEE Trans. Automat. Control, 45 (2000), pp. 1017–1022.

[48] A. Varga, *Computation of Kronecker-like forms of periodic matrix pairs*, in Proceedings of the 16th International Symposium on the Mathematical Theory of Networks and Systems, Leuven, Belgium, 2004, pp. 5–9.

[49] A. Varga, *Computation of* $\mathcal{L}_\infty$*-norm of linear discrete-time periodic systems*, in Proceedings of the 17th International Symposium on the Mathematical Theory of Networks and Systems (MTNS'06), Kyoto, Japan, 2006.

[50] A. Varga and P. Van Dooren, *Computing the zeros of periodic descriptor systems*, Systems Control Lett., 50 (2003), pp. 371–381.

[51] J. Vlach, K. Singhai, and M. Vlach, *Computer oriented formulation of equations and analysis of switched-capacitor networks*, IEEE Trans. Circuits Systems, 31 (1984), pp. 735–765.

[52] J. Xin, H. Kagiwada, A. Sano, H. Tsuj, and S. Yoshimoto, *Regularization approach for detection of cyclostationary signals in antenna array processing*, in Proceedings of the IFAC Symposium on System Identification, Vol. 2, 1997, pp. 529–534.

[53] V. A. Yakubovich and V. M. Starzhinskii, *Linear Differential Equations with Periodic Coefficients*, Wiley, New York, 1975.

[54] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice–Hall, Upper Saddle River, NJ, 1996.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 1007–1024

# TOWARDS STABLE MIXED PIVOTING STRATEGIES FOR THE SEQUENTIAL AND PARALLEL SOLUTION OF SPARSE SYMMETRIC INDEFINITE SYSTEMS*

IAIN S. DUFF† AND STÉPHANE PRALET‡

**Abstract.** We consider the direct solution of sparse symmetric indefinite matrices. We develop new pivoting strategies that combine numerical pivoting and perturbation techniques. Then an iterative refinement process uses our approximate factorization to compute a solution. We show that our pivoting strategies are numerically robust, that few steps of iterative refinement are required, and that the factorization is significantly faster than with previous methods. Furthermore, we propose original approaches that are designed for parallel distributed factorization. A key point of our parallel implementation is the cheap and reliable estimation of the growth factor. This estimation is based on an approximation of the off-diagonal entries and does not require any supplementary messages.

**Key words.** direct solver for sparse symmetric indefinite matrices, Gaussian elimination, parallel multifrontal methods, static pivoting, matrix perturbation

**AMS subject classifications.** 65F05, 65F50

**DOI.** 10.1137/050629598

## 1. Introduction.

We study pivoting strategies for computing the **LDL**$^{\mathbf{T}}$ factorization of a symmetric indefinite matrix, where $L$ is a lower triangular matrix and $D$ is a block diagonal matrix with $1\times1$ and $2\times2$ blocks. We consider direct methods based on a multifrontal technique, although most of our comments and analysis apply to other approaches for direct factorization.

Usually the factorization is computed in two phases. The analysis phase preprocesses the system of equations and is often based purely on matrix structure. The second phase performs the Gaussian elimination. If the numerical tests prevent the selection of some pivots chosen by the analysis, then the factorization can still proceed, but there will normally be an increase in both storage and work for the factorization compared to that required if no pivots are delayed. This effect can be particularly significant on augmented systems.

We would like to define "static pivoting" as any pivoting scheme that respects the data structures obtained from the analysis (so that static data structures can be used). However, this term is still a matter for debate within our community, so we therefore use the term [illegible] in the body of this paper. Such a scheme closely follows the pivot selection of the analysis to give generally lower fill-in and factorization times at the potential cost of worse accuracy in the factorization. In this paper we propose a new approach that is reliable in terms of numerical precision and memory estimation. Our new approach also improves the factorization time compared to approaches that perform usual numerical pivoting. The originality of our approach is to combine numerical pivoting and perturbation techniques and to propose a criterion for deciding between small $1\times1$ and small $2\times2$ pivots. When we

---

*Received by the editors April 19, 2005; accepted for publication (in revised form) by R. Nabben February 20, 2007; published electronically October 17, 2007.
http://www.siam.org/journals/simax/29-3/62959.html

†CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France, and Atlas Centre, Rutherford Appleton Laboratory, Oxon OX11 0QX, England (i.s.duff@rl.ac.uk).

‡CNRS-IRIT, ENSEEIHT, 2, rue Charles Camichel, BP 7122, F-31071 Toulouse Cedex 7, France (Stephane.Pralet@enseeiht.fr).

would select too small a pivot, we modify diagonal entries so that we compute the **LU** factorization of a perturbed matrix:

$$\mathbf{LU} = \mathbf{A} + \Delta + \mathbf{E},$$

where $\Delta$ is a diagonal matrix corresponding to the perturbed pivots and $\mathbf{E}$ corresponds to the rounding errors. We then use iterative refinement with this approximation and show that we almost always compute an accurate solution. We also find that delaying pivots is dangerous in the context of an approximate factorization. Moreover, we propose new pivoting strategies that are numerically robust on our representative test set and do not adversely affect the scalability of a parallel distributed factorization. They are based on estimations of growth factors and do not require any supplementary messages.

In our recent work [12], we developed preprocessing techniques to be used with sparse symmetric indefinite direct solvers which perform numerical pivoting:

- We showed how maximum weighted matching techniques can be used when the matrix is symmetric to effect a symmetric scaling. We have found this very beneficial over a wide range of problems. That is why we will systematically use scaling in this paper, although in some of the analysis we do not assume that the system will necessarily have been scaled.
- We also used symmetric weighted matching techniques to identify potential $2\times 2$ pivots. A major benefit of our work is that the analysis phase gives a better indication of the work and storage required by the subsequent factorization. Nevertheless the influence of our pivot preselection on the performance of the factorization depends very much on the nature of the matrix. It might increase the fill-in in the factors and the factorization time, for example. That is why we will not use this preprocessing in this paper (except in section 7, where we want to evaluate the combination of the preprocessing of [12] with the pivoting strategies presented in this paper).

Even if these preprocessing steps are sequential, they do not represent a significant overhead compared to the total analysis time and are negligible compared to the factorization time.

In section 2, we describe multifrontal solvers and parallel distributed multifrontal solvers. We also briefly describe pivoting strategies implemented in `SuperLU_DIST` [22] and PARDISO [27] and the numerical pivoting strategies that are often used in the context of sparse direct methods.

Section 3 presents our experimental environment. In section 4, we describe a new pivoting strategy that combines numerical pivoting and perturbation techniques and that is well designed for sequential factorization. We present experimental results for this strategy. In section 5, we show that combining restricted pivoting and delayed pivots severely affects the numerical quality of the factorization. Section 6 presents pivoting strategies that are particularly suited for parallel distributed solvers. In section 7, we study the influence of the preprocessings of [12] on our pivoting strategies. We present our conclusions in section 8.

### 1.1. Notation.

In the following, $s$ will be the function for the sign of a real number,

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0, \end{cases}$$

$\epsilon$ will denote the machine precision, $|| \; ||_2$ and $|| \; ||_\infty$ will denote the submultiplicative matrix norms, and $||A||_M$ will denote the norm $\max_{ij} |a_{ij}|$.

For each matrix or submatrix $A = (a_{ij})$, $|A| = (|a_{ij}|)$, $n$ will denote the order of $A$ and $nnz$ its number of nonzeros. $0 < u \leq 1$ and $\mu = \sqrt{\epsilon}$ will denote real numbers which will be used as thresholds in our pivoting strategies. In practice, we will use $u = 0.01$. $FSV$ and $PSV$ will denote fully summed variables and partially summed variables, respectively.

We summarize in Table 1.1 the main pivoting strategies that we develop in this paper.

Table 1.1

*Summary of our main pivoting strategies. The* `SEQ` *suffix means that the strategy is designed for a sequential code. The* `PAR` *suffix means that the strategy is designed for a parallel code.*

| Name | Pivoting strategy | Section |
|---|---|---|
| `numSEQ` | Numerical pivoting of Duff–Reid algorithm | 2.2 |
| `mixSEQ` | Numerical pivoting combined with perturbation techniques (restricted pivoting) | 4.1 |
| `numBPAR` | Basic restriction of Duff–Reid algorithm | 6.1 |
| `numEPAR` | Adaptation of Duff–Reid algorithm that uses estimations | 6.2 |
| `mixEPAR` | Combination of `numEPAR` and `mixSEQ` | 4.1 and 6.2 |

## 2. Symmetric indefinite multifrontal solvers and numerical pivoting.

**2.1. Multifrontal approach.** For an irreducible matrix, the [illegible] [13, 23] represents the order in which the matrix can be factorized, that is, in which the unknowns from the underlying linear system of equations can be chosen. (This tree is in the most general case a forest, but we will assume in our discussions, for the sake of clarity, that it is a connected tree. That is, the matrix is irreducible.) One central concept of the multifrontal approach [13] is to group (or [illegible]) columns with the same sparsity structure to create [illegible] or [illegible] [13, 24] in order to make use of efficient dense matrix kernels. The amalgamated elimination tree is called the [illegible].

$$
\begin{array}{rcc}
 & \text{fully summed columns} & \text{partially summed columns} \\
 & \downarrow & \downarrow \\
\text{fully summed rows} \rightarrow & F_{11} & F_{12} \\
\text{partially summed rows} \rightarrow & F_{21} & F_{22}
\end{array}
$$

Fig. 2.1. *A frontal matrix.*

When a node in the assembly tree is being processed, it assembles the contribution blocks from all its child nodes into its [illegible] (see Figure 2.1). In the symmetric case, pivots are usually chosen from the diagonal as discussed in section 2.2, and operations and storage are about half those of the general case. The pivotal variables from the [illegible] block, $F_{11}$, are eliminated, and the Schur complement matrix, $F_{22} - F_{21}F_{11}^{-1}F_{12}$, is computed. The contribution block is then sent to the parent node to be assembled. If some variables are not eliminated because of numerical issues, they are moved to the contribution block and sent to the parent node. The effect of this is that the computation $F'_{22} - F'_{21}{F'_{11}}^{-1}F'_{12}$ is performed, where $F'_{11}$ is a submatrix of $F_{11}$ of dimension the number of pivots selected at this stage. If this dimension is less than the order of $F_{11}$, then the difference represents the number of pivots [illegible] at this stage. Delayed pivots have the effect of causing extra fill-in and thus increase the memory and the number of operations for the factorization.

**2.2. Numerical pivoting.** In the unsymmetric case, at step $k$ of Gaussian elimination, the pivot $(p, q)$ is selected from the fully summed rows and columns. To limit the growth of the entries in the factors and thus to obtain a more accurate factorization, a test on the magnitude of the pivot is commonly used. $a_{pq}$ can be selected if and only if

$$|a_{pq}| \geq u \max_j |a_{pj}|, \tag{2.1}$$

where $u$ is a threshold parameter between 0 and 1. This criterion will ensure that the growth factor is limited to $1 + 1/u$.

In the symmetric indefinite case, we have to perform 1×1 and 2×2 pivoting if we want to keep the symmetry while maintaining stability. Pivot selection can be done using the Bunch–Parlett [6] or Bunch–Kaufman [5] algorithm, or a variation proposed by [4] that uses rook pivoting. In the context of sparse matrices, the criterion of the Duff–Reid algorithm that uses rook pivoting ([13], as modified in [14]) can be used to ensure a growth factor of less than $1 + 1/u$ at each step of Gaussian elimination. A $1 \times 1$ diagonal pivot can be selected if and only if it satisfies the inequality (2.1). A $2 \times 2$ pivot $P = \left(\begin{smallmatrix} a_{pp} & a_{pq} \\ a_{qp} & a_{qq} \end{smallmatrix}\right)$ can be selected if and only if it satisfies

$$|P^{-1}| \begin{pmatrix} \max_{k \neq p,q} |a_{pk}| \\ \max_{k \neq p,q} |a_{qk}| \end{pmatrix} \leq \begin{pmatrix} 1/u \\ 1/u \end{pmatrix}, \tag{2.2}$$

where $u$ is a threshold between 0 and $\frac{1}{2}$.

**2.3. Numerical pivoting and perturbation techniques.** Modifying the diagonal of the matrix instead of performing numerical pivoting was introduced by Stewart in [28]. In his approach, the magnitude of the perturbations can be quite large. Given a symmetric matrix $A$, not necessarily positive definite, approaches to compute a modified Cholesky factorization of $A + E$ with $E$ as small as possible have been developed in [7, 15, 16] (in a modified Cholesky $||E||_2 \geq -\min \lambda_i(A)$). The size of this perturbation is larger than we want, so we do not use modified Cholesky approaches. Furthermore, the approach of [7] is not adapted for sparse matrices because the pattern of $E$ can be significantly different from the pattern of $A$.

Our restricted pivoting strategy means that in our code pivoting is restricted to static data structures that have been predicted by the analysis phase. The factorization does not necessarily follow the pivot selected by the ordering exactly, and some slight variations are allowed. For example, in a multifrontal context, it is sufficient that the factorization decisions be compatible with the assembly tree (numerical pivoting is restricted to fully summed variables within a front, and delayed pivots are not allowed). A static approach was proposed by [21] in the context of **LU** factorization. The pivot order decided during the analysis is strictly followed. During Gaussian elimination when a forecast pivot is too small, [illegible]” are added to limit the growth of the factors in order to enhance the backward stability of the algorithm (see, for example, Theorem 11.4 of [20], which gives an upper bound of the residual after one iterative refinement). By [illegible]” we mean that the perturbed matrix is close enough to the original so that the factorization of the perturbed matrix is close to the factorization of the original matrix. If this is the case, then we would hope to have a cheap and sufficiently accurate solution using the computed factors.

`SuperLU_DIST` [22] adds small perturbations $\delta$ to the diagonal entries when the pivot $a_{ii}$ is too small so that $|a_{ii} + \delta| \geq \mu \; ||A||_M$. Note that the forecast large entries are permuted to the diagonal using maximum weighted matching techniques [11] and that no pivoting is performed within the supernodes.

At the same time as we were working on our pivoting algorithms (based on Duff–Reid pivoting), strategies based on the Bunch–Kaufman pivoting strategy using possible perturbations were developed in [27]. These developments are independent and give rise to different performance characteristics. The approach in [27] may degrade the precision of the solution and slow down the solution phase because of the increase in the number of iterative refinement steps [17]. Moreover, contrary to our pivoting strategies in section 6, the approach of [27] has not been designed explicitly for a parallel distributed environment. Finally, the approach in [27] does not use estimations of the growth factor to control the accuracy of the factorization [20].

**2.4. Parallel distributed approaches.** This section discusses the parallelism that is exploited by distributed multifrontal solvers, focusing on the `MUMPS` approach and using the terminology from that work [1, 2].

A pair of nodes in the assembly tree, where neither is an ancestor of the other, can be factorized independently from each other, in any order or in parallel. Consequently, independent branches of the assembly tree can be processed in parallel, and we refer to this as *tree parallelism* or *type 1 parallelism*. It is obvious that, in general, tree parallelism can be exploited more efficiently in the lower part of the assembly tree than near the root node. Additional parallelism is then created using distributed memory versions of blocked algorithms to factorize the frontal matrices (see, for example, [2, 9]).

The lower triangular part of the frontal matrix is partitioned, and each part of it is assigned to a different process. The so-called *master process* is responsible for the diagonal block of fully summed variables and also decides which processes (the so-called *slave processes*) will be involved in the parallel activity associated with this node. We refer to this as *type 2 parallelism* and call the nodes involved *type 2 nodes*. After computing its part of the contribution block, a slave will communicate with the master and the slaves of the parent node.

In order to have an efficient factorization, asynchronous schemes are used. There is no synchronization between a master and its slaves (who may be involved in another task or even be the master of another type 2 node). Hence the master has to select pivots without the knowledge of the part of fully summed columns stored on its slaves. This problem will be addressed by our pivoting strategies presented in section 6. An alternative could be to overload the master process with the work associated with the complete fully summed columns (diagonal plus off-diagonal part), but obviously this would generate workload and memory imbalances and degrade the scalability of the approach.

**3. Experimental environment.** Our experiments are conducted on one node of a COMPAQ Alpha Server SC45 at CERFACS. There are four GBytes of memory shared between four EV68 processors per node, and we disable three of the processors so that we can use all the memory of the node with the remaining single processor. We use the Fortran 90 compiler, f90 version 5.5 with the -O option. If the factorization needs more than 4 GBytes or requires more than 30 minutes CPU time, we consider that it is not successful.

We conduct our experiments on a number of challenging and sometimes badly conditioned test problems. The matrices are available from ftp.numerical.rl.ac.uk/

Table 3.1

*Symmetric indefinite matrices.* $\lambda+$ *: number of positive eigenvalues.* $\lambda-$ *: number of negative eigenvalues.*

| Matrix | $n$ | $nnz$ | $\lambda+$ | $\lambda-$ | Origin |
|---|---|---|---|---|---|
| BRAINPC2 | 27607 | 96601 | 13807 | 13800 | Biological model (CUTEr) |
| BRATU3D | 27792 | 88627 | 15625 | 12167 | 3D Bratu problem on the unit cube (CUTEr) |
| CONT-201 | 80595 | 239596 | 40397 | 40198 | KKT matrix–Convex QP (M2) |
| CONT-300 | 180895 | 562496 | 90597 | 90298 | KKT matrix–Convex QP (M2) |
| crystk02 | 13965 | 491274 | 13964 | 1 | Stiffness matrix–crystal free vibration (UF) |
| crystk03 | 24696 | 887937 | 24695 | 1 | Stiffness matrix–crystal free vibration (UF) |
| cvxqp3 | 17500 | 62481 | 10000 | 7500 | Convex QP (CUTEr) |
| mario001 | 38434 | 114643 | 23130 | 15304 | Stokes equation (MA) |
| NCVXQP1 | 12111 | 40537 | 7111 | 5000 | KKT matrix–nonconvex QP (CUTEr) |
| NCVXQP5 | 62500 | 237483 | 28534 | 33966 | KKT matrix–nonconvex QP (CUTEr) |
| NCVXQP7 | 87500 | 312481 | 37500 | 50000 | KKT matrix–nonconvex QP (CUTEr) |
| SIT100 | 10262 | 34094 | 7143 | 3119 | Straz pod Ralskem mine model (MT) |
| stokes128 | 49666 | 295938 | 33281 | 16385 | Stokes equation (MA) |
| stokes64 | 12546 | 74242 | 8449 | 4097 | Stokes equation (AW) |

pub/matrices/symmetric/indef/ and most of them (all except the cvxqp3 matrix) are a subset of the matrices collected by [19] for testing symmetric sparse solvers. To select our matrices we ran our pivoting strategies on the matrices from [19] and kept the difficult matrices that illustrate the characteristics of our pivoting strategies. (We use "difficult" in the sense that they often required iterative refinement steps to get a small residual.)

Some of the matrices come from the Maros and Meszanos quadratic programming collection (M2) [25], the CUTEr optimization test set (CUTEr) [18], and the University of Florida collection (UF) [8]. Some problems were generated by Andy Wathen (AW), Mario Arioli (MA), and Miroslav Tuma (MT). These problems are described in Table 3.1. These test matrices correspond to augmented matrices of the form

$$\mathcal{K}_{H,A} = \begin{pmatrix} H & A \\ A^T & 0 \end{pmatrix}.$$

In all our experiments we use random right-hand sides. To perform an error analysis of the solution we compute the sparse componentwise backward error using the theory and measure developed by [3]. The scaled residual of the $i$th equation is

$$\Delta_i = \frac{|r_i|}{(|A||x| + |b|)_i},$$

where $r = b - Ax$ and $x$ is the computed solution, except if the denominator is too small (in our code the threshold for this is $1000 \times \epsilon$). In this case, we use

$$\Delta_i = \frac{|r_i|}{((|A||x|)_i + ||A_i||_\infty ||x||_\infty)_i},$$

where $A_i$ represents the $i$th row of $A$. We apply iterative refinement in all our approaches. At each step $k$ of the iterative refinement, we compute the current backward error $berr^{(k)} \;=\; \max_i \Delta_i$.

We also compute the normwise backward error

$$nberr \;=\; ||\mathbf{Ax} - \mathbf{b}||_\infty / (||\mathbf{A}||_\infty ||\mathbf{x}||_\infty + ||\mathbf{b}||_\infty)$$

and display it in Table 4.1 below to show that our approach is competitive with that presented in [27].

During our performance analysis, we will report the factorization time and the solution time separately because the total time really depends on the application. For example, if multiple solutions are required with the same matrix (as is often the case in optimization), the solution time may become dominant.

We will use our pivoting strategies with a symmetric multifrontal code, `MA57` Version 3.0.0 [10], on a challenging test set (see section 3). Note that `MA57` is one of the best sparse symmetric indefinite sequential direct solvers [17], and thus the advances presented here are advances on an already very competitive code. Unless otherwise stated, we use the METIS nested-dissection ordering during the analysis, and we symmetrically scale the matrix with our symmetric scaling [12]. This makes all entries smaller than 1 in magnitude, and the columns of the scaled matrix can be permuted so that it has entries of magnitude 1 on the diagonal. Thus in our experiments and in our criteria for selecting pivots we will have $||A||_M = 1$. Nevertheless, we will keep this term because in practice the matrix may be differently scaled or not scaled at all (this is an option for the user through the `MA57` interface). In these cases we observed that we obtain better solutions if we include the $||A||_M$ term in our criteria.

Although `MA57` is a sequential code, we will use it to simulate the parallel behavior of a multifrontal solver like `MUMPS` [1, 2].

## 4. Mixing numerical pivoting and perturbation techniques.

**4.1. Algorithm.** In this section, we present an approach which combines numerical checking for stability with perturbations. This approach will be referred to as [illegible]. We decided to use only $1 \times 1$ perturbations because they are easier to implement, and we want to clearly identify the impact of this mixed approach. Some promising experiments with $2 \times 2$ perturbations can be found in [26], but they were applied in a different context; the analysis fixes the 1×1 and 2×2 pivots, and appropriate perturbations were applied during the factorization. In this present paper, the pivots are dynamically chosen, and we have found that the use of $2 \times 2$ perturbations does not give a significant improvement. In our algorithm, we perturb the original entries only in extreme cases where pivots are very small in magnitude. That is why the additional use of $2 \times 2$ perturbations does not affect the precision of the solution.

Let us consider a frontal matrix from the elimination tree. It contains two kinds of variables, the $FSV$, which correspond to the pivot block that we want to eliminate, and the $PSV$, on which the Schur complement will be computed.

Our mixed approach is based on two phases. In the first phase, we perform numerical pivoting in the block of fully summed variables until no remaining variables satisfy the numerical criterion. In the second phase, we eliminate the remaining fully summed variables, adding $1 \times 1$ perturbations if necessary.

Our pivot selection is more precisely defined as follows. Using (2.1) and (2.2), we define

$$g_1(i) = \frac{\max_{k \neq i} |a_{ik}|}{|a_{ii}|}, \tag{4.1}$$

and

$$g_2(i,j) = \left\| |P^{-1}| \begin{pmatrix} \max_{k \neq i,j} |a_{ik}| \\ \max_{k \neq i,j} |a_{jk}| \end{pmatrix} \right\|_\infty . \tag{4.2}$$

During the first phase of Algorithm 1 (the usual Duff–Reid algorithm), a 1×1 pivot $a_{ii}$ is considered to be stable if and only if

$$g_1(i) \leq 1/u, \tag{4.3}$$

and a 2×2 pivot $P$ is considered to be stable if and only if

$$g_2(i,j) \leq 1/u. \tag{4.4}$$

**Algorithm 1** Numerical pivot selection combined with perturbation techniques.

```
Phase 1: Eliminate as many 1×1 and 2×2 pivots as possible which satisfy inequalities (4.3) and (4.4), respectively, using the Duff–Reid algorithm with threshold $u$.
Phase 2:
while $FSV \neq \emptyset$ do
  Let $i \in FSV$.
  if $\#FSV = 1$ then
    if $|a_{ii}| < \mu\ ||A||_M$ then $a_{ii} = s(a_{ii})\mu\ ||A||_M$
    Perform elimination using $i$ as a 1×1 pivot.
    return
  end if
  Let $j = \arg\max_{k \in FSV \setminus \{i\}} |a_{ik}|$ and $P$ be the 2×2 block associated with $i$ and $j$.
  Choose between 1×1 or 2×2 pivoting:
  if $\min\{g_1(i), g_2(i,j)\} < 1/\mu$ then /* Case 1: growth factor comparison */
    if $g_2 < g_1$ then
      Perform elimination using $(i,j)$ as a 2×2 pivot.
    else
      Perform elimination using $i$ as a 1×1 pivot.
    end if
  else if $\min\{1/|a_{ii}|, ||P^{-1}||_\infty\} < 1/(\mu\ ||A||_M)$ then /* Case 2: pivot size comparison */
    if $1/|a_{ii}| > ||P^{-1}||_\infty$ then
      Perform elimination using $(i,j)$ as a 2×2 pivot.
    else
      Perform elimination using $i$ as a 1×1 pivot.
    end if
  else /* Case 3: diagonal 1×1 perturbation */
    $a_{ii} = s(a_{ii})\mu\ ||A||_M$
    Perform elimination using $i$ as a 1×1 pivot.
  end if
end while
```

During the second phase we use the threshold $\mu$. We perturb the diagonal of the matrix if the pivot is too small with respect to the initial values in $A$ (smaller than $\mu\ ||A||_M$). We tried different thresholds for $\mu$ in [26] and compared the precision of the solution after applying iterative refinement. We remark that choosing $10^{-7} \leq \mu \leq 10^{-10}$ seems to be a good compromise between small perturbations and small growth factors (larger $\mu$ values would give a more stable factorization but at the cost of obtaining the factorization of a matrix further from the original). Moreover, we observed that different values of $\mu$ in the range $[10^{-7}, 10^{-10}]$ lead to similar behavior in terms of number of iterations and precision of the solution. In our experiments $\mu$ is fixed at $\sqrt{\epsilon} \approx 10^{-8}$.

The choice between a 1×1 and a 2×2 pivot is done in three stages. First (Case 1 of Algorithm 1), if we can eliminate a pivot and ensure a growth factor lower than $1 + 1/\mu$, then we select the one with the lower growth factor. Second (Case 2 of Algorithm 1), if we cannot ensure a growth factor lower than $1+1/\mu$, then we compare the quantities $1/|a_{ii}|$ and $||P^{-1}||_\infty$. This second comparison is guided by the growth factor that would appear if we suppose that the largest off-diagonal entry is bounded by $||A||_M$. Finally, if no pivot can be chosen, a perturbed 1×1 pivot is selected (Case 3 of Algorithm 1).

Table 4.1
*Componentwise and normwise (columns nberr) backward error and number of tiny pivots.*

| Matrix | `numSEQ` pivoting strategy | | | `mixSEQ` pivoting strategy | | | | |
|---|---|---|---|---|---|---|---|---|
| | it. 0 | it. 1 | *nberr* | it. 0 | it. 1 | it. 2 | *nberr* | Tiny |
| BRAINPC2 | 8.1e-14 | 2.2e-16 | 2.7e-16 | 2.9e-06 | 3.2e-12 | 2.0e-15 | 1.2e-15 | 12932 |
| BRATU3D | 1.6e-09 | 2.9e-16 | 1.5e-16 | 1.6e-05 | 6.9e-12 | 2.9e-16 | 1.5e-16 | 8429 |
| CONT-201 | 1.2e-10 | 2.6e-16 | 1.1e-16 | 2.7e-05 | 4.0e-08 | 3.2e-08 | 2.3e-09 | 27470 |
| CONT-300 | 1.4e-10 | 2.3e-16 | 1.4e-16 | 6.4e-05 | 8.6e-08 | 8.6e-08 | 2.2e-09 | 67864 |
| crystk02 | 5.0e-16 | 4.3e-16 | 1.1e-16 | 5.0e-16 | 4.2e-16 | 4.3e-16 | 1.1e-16 | 0 |
| crystk03 | 7.7e-16 | 4.3e-16 | 1.7e-16 | 7.7e-16 | 3.9e-16 | 4.3e-16 | 1.7e-16 | 0 |
| cvxqp3 | 3.2e-10 | 3.9e-16 | 1.4e-16 | 1.3e-03 | 4.4e-12 | 4.4e-16 | 1.4e-16 | 6277 |
| mario001 | 3.0e-14 | 2.5e-16 | 6.6e-17 | 1.6e-07 | 4.8e-12 | 3.3e-16 | 6.6e-17 | 10305 |
| NCVXQP1 | 1.8e-03 | 2.2e-16 | 2.6e-17 | 9.9e-01 | 1.1e-04 | 1.8e-11 | 3.7e-17 | 3619 |
| NCVXQP5 | 6.3e-10 | 3.1e-16 | 1.1e-16 | 5.7e-05 | 1.6e-08 | 5.3e-12 | 1.4e-14 | 8402 |
| NCVXQP7 | 4.6e-09 | 3.5e-16 | 1.8e-16 | 2.0e-02 | 1.5e-10 | 8.4e-16 | 1.8e-16 | 31043 |
| SIT100 | 1.0e-14 | 1.5e-15 | 1.6e-15 | 3.0e-07 | 6.7e-14 | 2.1e-16 | 1.1e-16 | 1388 |
| stokes128 | 3.8e-14 | 3.4e-15 | 1.3e-15 | 6.0e-07 | 3.0e-12 | 4.0e-15 | 6.0e-16 | 12738 |
| stokes64 | 9.2e-15 | 6.5e-15 | 8.1e-15 | 4.4e-07 | 1.2e-12 | 7.9e-15 | 1.1e-15 | 3106 |

We also tried to smooth the transition between the first and the second phase. Before the second phase, we performed numerical pivoting with smaller values of $u$. More precisely we defined a parameter $u_{min}$. While $u$ is larger than $u_{min}$, we decrease the $u$ value (for example, $u = u/10$) and restart phase 1. We did not observe gains from such an approach and thus prefer to focus on the simple version of our pivoting strategies (Algorithm 1) and to keep $u$ fixed at 0.01.

**4.2. Experimental results.** In this section, we discuss the influence of the pivoting scheme described in section 4.1. It will be referred to as the `mixSEQ` algorithm because checking the stability over the partially summed rows is not well designed for existing parallel implementations (see section 2.4).

In the rest of the paper, `numSEQ` will refer to numerical pivoting strategy of Duff and Reid [13]. When we perturb a 1×1 pivot (Case 3 of Algorithm 1) this pivot is called a [illegible].

Table 4.1 compares the precision of the solution for an $\mathbf{LDL^T}$ factorization with numerical pivoting and an $\mathbf{LDL^T}$ factorization with the `mixSEQ` pivoting strategy. Because of numerical pivoting, no iterative refinement is needed for a backward error smaller than $\sqrt{\epsilon}$, whereas with `mixSEQ` the backward error without any iterative refinement is often larger than $\sqrt{\epsilon}$. Thus we advise performing one or two steps of iterative refinement when using the `mixSEQ` strategy. This slight degradation of precision is due to the tiny pivots. On average, the `mixSEQ` strategy needs one iteration more to get the same precision as the `numSEQ` strategy.

Although the number of tiny pivots can be very large (see last column of Table 4.1), our `mixSEQ` approach generally succeeds in reducing the backward error. The CONT-* matrices are the only ones for which iterative refinement does not converge to the machine precision, even with several iterations.

Columns *nberr* of Table 4.1 show that our approach is competitive with and sometimes better than the approach presented in [27] (for example, on crystk* matrices the PARDISO normwise backward error is between $10^{-13}$ and $10^{-12}$, and on stokes* matrices it is between $10^{-12}$ and $10^{-10}$). In the rest of the paper we will concentrate on sparse componentwise backward errors since they give us more information about the quality of the solution [3].

Table 4.2 shows the main advantage of using restricted pivoting: the `mixSEQ` factorization is always faster. Delaying pivots increases the number of operations

Table 4.2

*Factorization and solution time (in seconds), number of delayed pivots, and size of the factors (in thousands of reals).* `numSEQ`*: Duff–Reid pivoting strategy.* `mixSEQ`*: combination of numerical pivoting and perturbation techniques. * means that the* `mixSEQ` *numerical quality is not similar to the* `numSEQ` *quality. Number delayed: number of delayed pivots. MV time: matrix-vector multiplication time (in seconds).*

| Matrix | Number delayed `numSEQ` | Factorization time `numSEQ` | Factorization time `mixSEQ` | Time for forward and backward substitution `numSEQ` | Time for forward and backward substitution `mixSEQ` | MV time | Size of the factors `numSEQ` | Size of the factors `mixSEQ` |
|---|---|---|---|---|---|---|---|---|
| BRAINPC2 | 14267 | 0.18 | 0.11 | 0.018 | 0.014 | 0.003 | 656 | 322 |
| BRATU3D | 90052 | 34.2 | 9.24 | 0.255 | 0.125 | 0.003 | 11484 | 5569 |
| CONT-201 | 71296 | 5.51 | 1.94* | 0.195 | 0.127* | 0.008 | 8820 | 4304 |
| CONT-300 | 183306 | 21.1 | 6.08* | 0.547 | 0.306* | 0.033 | 23838 | 10714 |
| cvxqp3 | 30519 | 9.73 | 3.08 | 0.099 | 0.048 | 0.002 | 4740 | 2301 |
| mario001 | 15463 | 0.28 | 0.23 | 0.024 | 0.022 | 0.008 | 817 | 575 |
| NCVXQP1 | 12463 | 2.69 | 1.29 | 0.039 | 0.024 | 0.001 | 2235 | 1327 |
| NCVXQP5 | 16703 | 25.7 | 23.0 | 0.326 | 0.279 | 0.015 | 13365 | 11205 |
| NCVXQP7 | 195973 | 195. | 71.6 | 0.874 | 0.498 | 0.021 | 37683 | 19367 |
| SIT100 | 2710 | 0.13 | 0.11 | 0.007 | 0.004 | 0.001 | 483 | 417 |
| stokes128 | 18056 | 1.14 | 1.06 | 0.096 | 0.076 | 0.016 | 3437 | 2753 |
| stokes64 | 4292 | 0.33 | 0.29 | 0.012 | 0.013 | 0.002 | 736 | 577 |

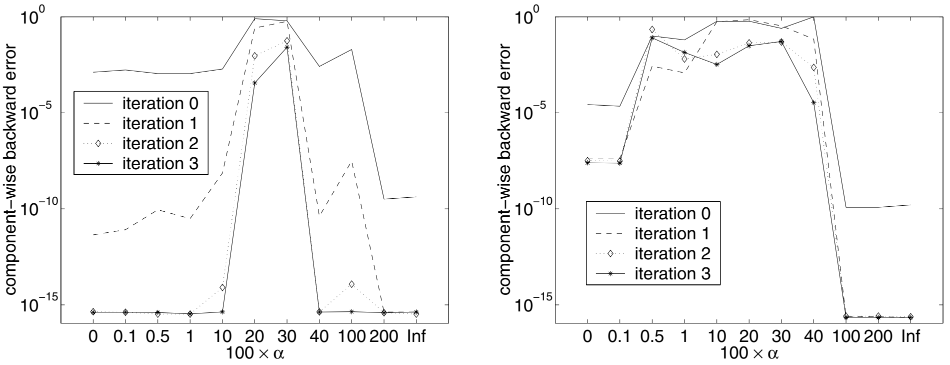
and thus tends to slow down the factorization phase. As `mixSEQ` generates sparser factors, it decreases the time for backward and forward substitution. Nevertheless the solution phase is often more costly with the `mixSEQ` strategy because it requires more iterative refinement steps and thus more backward and forward substitutions and matrix-vector multiplications.

**5. Combination of `mixSEQ` and delayed pivots.** A quite natural idea to improve the precision of the solution is to allow some delayed pivots up to a certain predetermined limit. This criterion could be based, for example, on the memory increase. We developed an approach that proceeds as follows:

- first performs pivoting using the Duff–Reid algorithm with a threshold $u$ until $\alpha \times n$ pivots have been delayed, where $\alpha$ is a real parameter (each time a pivot is delayed to its parent, one is added to the count for delayed pivots, and so there may be more than $n$ delayed pivots), and
- second uses Algorithm 1.

We show in Figure 5.1 the precision of the solution while increasing the number of delayed pivots allowed. The approach with $\alpha = 0$ is equivalent to the `mixSEQ` strategy, and $\alpha = \infty$ ($Inf$ in Figure 5.1) corresponds to the `numSEQ` strategy. We see that combining delayed pivots and `mixSEQ` pivoting may be dangerous: this combination often requires more steps of iterative refinement to obtain a small residual. We think that the poor convergence of the iterative refinement process and sometimes the degradation in the precision is due to the accumulation of both rounding errors and perturbations. When delayed pivots are not allowed, the diagonal perturbations and their influence on rounding errors are localized to the contribution blocks predicted by the analysis. If we allow delayed pivots, some pivots can be postponed, and it is possible that they remain unacceptable in ancestor nodes until we switch to `mixSEQ` mode. In that case when a delayed pivot (possibly postponed over several generations) is still small, it is perturbed. This elimination then contaminates a larger contribution block than if it had been eliminated at an earlier node.

Figure 5.1 also illustrates that it is difficult to predict the behavior of the precision. For example, there is a small window, $10 \leq 100\alpha \leq 40$, in which our algorithm does



(a) cvxqp3 (b) CONT-201

FIG. 5.1. *Influence of the number of delayed pivots on the precision of the solution.*

not return an accurate solution for cvxqp3. The failure window is completely different for CONT-201, $0.1 \leq 100\alpha \leq 100$.

Finally, let us mention that we tried more sophisticated rules (based on the topology of the assembly tree, the characteristics of the parent node, etc.) to decide how to delay the elimination of a variable, and we note that all these strategies were very unstable.

## 6. Pivoting strategies in parallel distributed environments.

**6.1. Limiting the areas for pivot checking.** The diagonal block of the fully summed part of a type 2 node is stored on a single processor, the master (see section 2.4). Furthermore, the master does not have local access to the other rows of the front, which are sent directly from the slaves of its child nodes to its own slaves. To avoid extra communications and, even worse, synchronizations we modify the quantities $g_1$ and $g_2$ of (4.1) and (4.2). Moreover, because we do not have access to all of the fully summed columns, we always suppose that the largest off-diagonal entry is greater than $\mu||A||_M$. Thus, for every type 2 node we define

$$g_1(i) = \frac{\max\{\max_{k\in FSV\setminus\{i\}} |a_{ik}|, \mu\ ||A||_M\}}{|a_{ii}|}, \tag{6.1}$$

and

$$g_2(i,j) = \left\| |P^{-1}| \begin{pmatrix} \max\{\max_{k\in FSV\setminus\{i,j\}} |a_{ik}|, \mu\ ||A||_M\} \\ \max\{\max_{k\in FSV\setminus\{i,j\}} |a_{jk}|, \mu\ ||A||_M\} \end{pmatrix} \right\|_\infty . \tag{6.2}$$

These quantities are then used to select pivots in type 2 nodes in both the Duff–Reid algorithm and in Algorithm 1. This **B**asic **PAR**allel strategy will be referred to as `BPAR`. When it is used with the Duff–Reid algorithm it will be denoted by `numBPAR`.

The `numBPAR` strategy is more friendly than `numSEQ` for the parallel distributed implementation of `MUMPS` [1] and more generally for other distributed solvers, for example, `SuperLU_DIST` [22]. Note that a similar strategy is already used in the symmetric indefinite code of `MUMPS`.

**6.2. Cheap estimation of growth factors.** We will see in section 6.3 that the `numBPAR` approach is not robust. In this section, we propose a better approximation of the off-diagonal information that does not limit the scalability and that will significantly improve the numerical robustness of the factorization. The main principle of our approach is to send information from the slaves of a node to the master of its parent when they send information related to their contribution block.

Let $c$ be a child node and $p$ be its parent. Let $i$ be a fully summed variable of node $p$. For each slave $s$ of node $c$, we define $m_i(s)$ the maximum entry (in magnitude) in row/column $i$ that is stored in the contribution block of process $s$. $s$ will send to the master of $p$ the quantities $m_i(s)$ for each fully summed variable of the node $p$. The master of node $p$ will approximate the maximum entry in each fully summed column using the maximum quantity that it has received from the slaves of its child nodes. It is only an approximation because the child contributions are not summed by the master of $p$, and no account is taken of numerical growth as the eliminations at node $p$ are performed. Hence, while receiving information about the maximum off-diagonal entries, $m_i(s)$, the master of the parent computes the quantities $M_i$:

$$M_i = \max\left\{ \max_{c \text{ child of } p} \left\{ \max_{s \text{ slave of } c} \{m_i(s)\} \right\}, \max_{k\in PSV} |a_{ik}^{(0)}| \right\}, \tag{6.3}$$

where $a^{(0)}$ denotes the original entries of $\mathbf{A}$ that are assembled at the parent node and that can be easily predicted before the factorization.

For each pivoting strategy (numerical pivoting or restricted pivoting), the $g_1$ and $g_2$ quantities of (6.1) and (6.2) are modified. For each type 2 node we define

$$g_1(i) = \frac{\max\{\max_{k\in FSV} |a_{ik}|, M_i, \mu \, ||A||_M\}}{|a_{ii}|} \tag{6.4}$$

and

$$g_2(i,j) = \left\| |P^{-1}| \begin{pmatrix} \max\{\max_{k\in FSV} |a_{ik}|, M_i, \mu \, ||A||_M\} \\ \max\{\max_{k\in FSV} |a_{ik}|, M_j, \mu \, ||A||_M\} \end{pmatrix} \right\|_\infty . \tag{6.5}$$

These quantities are then used to adapt either the Duff–Reid algorithm or Algorithm 1 on type 2 nodes to parallel distributed environments. These **E**stimations of off-diagonal entries in a **PAR**allel framework will be referred to as `EPAR`. The adaptation of Duff–Reid **NUM**erical pivoting strategy will be referred to as `numEPAR`, and the adaptation of the **MIX**ted approach of Algorithm 1 will be referred to as `mixEPAR`.

The areas checked for the computation of the $m_i(s)$ quantities are illustrated in Figure 6.1. Each slave accesses the shaded areas to compute its $m_i$ quantities for each fully summed variable of the parent node. Then it communicates them to the father while sending the black blocks of its contributions. The other parts of the contribution blocks (shaded and blank) are sent directly to the slaves of the parent node.

**6.3. Experimental results.** The precision of the solution with our parallel approaches will be affected by the choice of the type 2 nodes because of the issues discussed in section 2.4. In `MUMPS` it depends on the characteristics of the nodes in the assembly tree (front size, number of fully summed variables) and the number of MPI processes. Indeed, the number of type 2 nodes increases with the number of processes in order to have more parallelism in the upper part of the assembly tree. In our simulations, we consider that a node is a type 2 node if it is large enough (in practice, if $\#PSV > 400$) and if it is neither a leaf node nor the root node. We remark that with this choice we simulate the behavior of `MUMPS` on hundreds of processes.

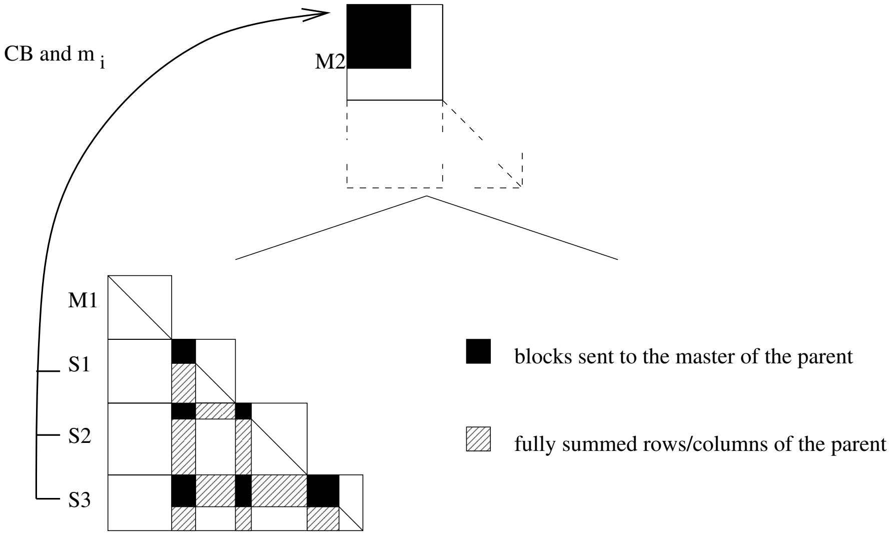

CB and m i
M2
M1
S1
S2
S3
blocks sent to the master of the parent
fully summed rows/columns of the parent

FIG. 6.1. *Illustration of the areas accessed to estimate the $m_i(s)$ quantities and of the blocks that are sent from a slave of a child to the master of the parent.*

TABLE 6.1
*Componentwise backward error of strategies with numerical pivoting.* `numSEQ`: *sequential approach.* `numBPAR`: *basic parallel approach.* `numEPAR`: *parallel approach using estimations.*

| | Iteration 0 | | | Iteration 1 | | |
|---|---|---|---|---|---|---|
| Matrix | `numSEQ` | `numBPAR` | `numEPAR` | `numSEQ` | `numBPAR` | `numEPAR` |
| BRAINPC2 | 8.1e-14 | 8.1e-14 | 8.1e-14 | 2.2e-16 | 6.1e-16 | 6.1e-16 |
| BRATU3D | 1.6e-09 | 1.0e+00 | 1.3e-08 | 2.9e-16 | 1.0e+00 | 2.6e-16 |
| CONT-201 | 1.2e-10 | 1.0e+00 | 1.6e-10 | 2.6e-16 | 9.9e-01 | 2.1e-16 |
| CONT-300 | 1.4e-10 | 1.0e+00 | 1.5e-10 | 2.3e-16 | 1.0e+00 | 2.8e-16 |
| cvxqp3 | 3.2e-10 | 1.0e+00 | 4.2e-10 | 3.9e-16 | 1.0e+00 | 4.2e-16 |
| mario001 | 3.0e-14 | 3.0e-14 | 3.0e-14 | 2.5e-16 | 2.0e-16 | 2.0e-16 |
| NCVXQP1 | 1.8e-03 | 1.0e+00 | 9.9e-04 | 2.2e-16 | 1.0e+00 | 2.3e-16 |
| NCVXQP5 | 6.3e-10 | 1.0e+00 | 4.1e-09 | 3.1e-16 | 1.0e+00 | 3.3e-16 |
| NCVXQP7 | 4.6e-09 | 1.0e+00 | 3.2e-09 | 3.5e-16 | 1.0e+00 | 3.7e-16 |
| SIT100 | 1.0e-14 | 1.0e-14 | 1.0e-14 | 1.5e-15 | 1.7e-14 | 1.7e-14 |
| stokes128 | 3.8e-14 | 3.8e-14 | 3.8e-14 | 3.4e-15 | 8.0e-15 | 8.0e-15 |
| stokes64 | 9.2e-15 | 9.2e-15 | 9.2e-15 | 6.5e-15 | 1.2e-14 | 1.2e-14 |

### 6.3.1. Parallel adaptation of Duff–Reid algorithm.

Table 6.1 shows that the basic parallel adaptation of the Duff–Reid algorithm, `numBPAR`, is not robust. We observe numerical failures on the CONT-201, CONT-300, and BRATU3D matrices. Note that we also observe these failures with `MUMPS` with more than four processes. Furthermore, there is a significant degradation of the precision compared to the `numSEQ` approach. In contrast, `numEPAR` is robust (no numerical failures), and it returns backward errors similar to the `numSEQ` strategy.

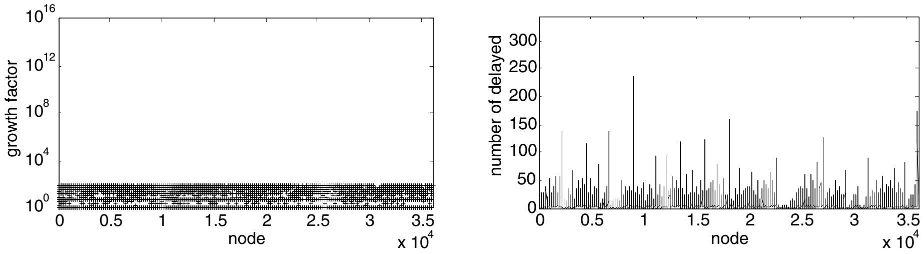
Table 6.2 compares the size of the factors and the number of delayed pivots between the `numSEQ`, `numBPAR`, and `numEPAR` pivoting strategies. We focus on the BRATU3D and CONT-* matrices because they reveal the weaknesses of the `numBPAR` strategy.

First we observe that `numSEQ` and `numEPAR` have a similar number of entries in the factors and number of delayed pivots. This supports the idea that the `numEPAR`

Table 6.2

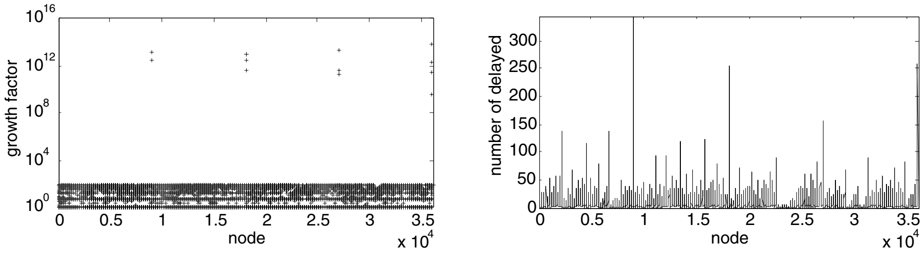
*Size of the factors and number of delayed pivots with different numerical pivoting strategies.*

| | Size of the factors | | | Number of delayed pivots | | |
|---|---|---|---|---|---|---|
| Matrix | `numSEQ` | `numBPAR` | `numEPAR` | `numSEQ` | `numBPAR` | `numEPAR` |
| BRATU3D | 11484379 | 9672751 | 11249260 | 90052 | 49205 | 87167 |
| CONT-201 | 8820367 | 8918700 | 8829464 | 71296 | 71415 | 71389 |
| CONT-300 | 23838606 | 23595744 | 23928663 | 183306 | 182422 | 183641 |



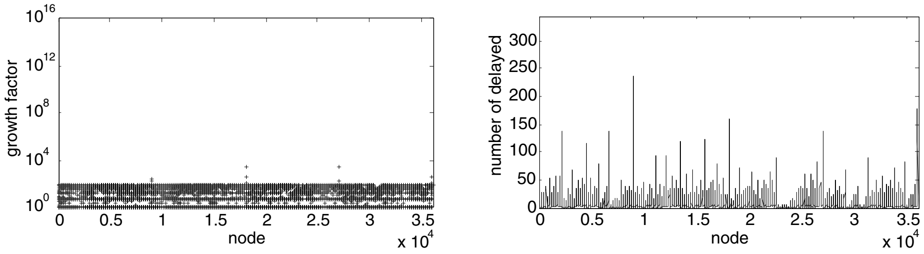
(a) Growth factor on CONT-201 using `numSEQ` pivoting strategy.

(b) Number of delayed pivots on CONT-201 using `numSEQ` pivoting strategy.



(c) Growth factor on CONT-201 using `numBPAR` pivoting strategy.

(d) Number of delayed pivots on CONT-201 using `numBPAR` pivoting strategy.



(e) Growth factor on CONT-201 using `numEPAR` pivoting strategy.

(f) Number of delayed pivots on CONT-201 using `numEPAR` pivoting strategy.

Fig. 6.2. *Influence of the pivoting strategy on the growth factor and the number of delayed pivots on CONT*-201.

strategy takes a good numerical decision even if it has only an approximate view of the off-diagonal entries. For each node of the assembly tree, Figure 6.2 represents the number of pivots that are delayed (right-hand side) and the maximum of the quantities

Table 6.3

*Componentwise backward error of strategies with combination of numerical pivoting and perturbation techniques.* `mixSEQ`*: sequential approach.* `mixEPAR`*: a parallel approach using estimations of* (6.3) *and pivoting Algorithm* 1.

| | Iteration 0 | | Iteration 1 | | Iteration 2 | |
|---|---|---|---|---|---|---|
| Matrix | `mixSEQ` | `mixEPAR` | `mixSEQ` | `mixEPAR` | `mixSEQ` | `mixEPAR` |
| BRAINPC2 | 2.9e-06 | 2.9e-06 | 3.2e-12 | 3.2e-12 | 2.0e-15 | 2.0e-15 |
| BRATU3D | 1.6e-05 | 8.2e-06 | 6.9e-12 | 4.9e-11 | 2.9e-16 | 3.8e-16 |
| CONT-201 | 2.7e-05 | 2.1e-05 | 4.0e-08 | 4.0e-08 | 3.2e-08 | 3.2e-08 |
| CONT-300 | 6.4e-05 | 2.6e-05 | 8.6e-08 | 8.6e-08 | 8.6e-08 | 1.7e-07 |
| cvxqp3 | 1.3e-03 | 1.4e-03 | 4.4e-12 | 8.8e-12 | 4.4e-16 | 3.5e-16 |
| mario001 | 1.6e-07 | 1.6e-07 | 4.8e-12 | 4.8e-12 | 3.3e-16 | 3.3e-16 |
| NCVXQP1 | 9.9e-01 | 9.9e-01 | 1.1e-04 | 6.8e-05 | 1.8e-11 | 1.7e-11 |
| NCVXQP5 | 5.7e-05 | 5.9e-05 | 1.6e-08 | 1.7e-08 | 5.3e-12 | 7.7e-12 |
| NCVXQP7 | 2.0e-02 | 1.3e-02 | 1.5e-10 | 2.0e-11 | 8.4e-16 | 3.4e-16 |
| SIT100 | 3.0e-07 | 3.0e-07 | 6.7e-14 | 6.7e-14 | 2.1e-16 | 2.1e-16 |
| stokes128 | 6.0e-07 | 6.0e-07 | 3.0e-12 | 3.0e-12 | 4.0e-15 | 4.0e-15 |
| stokes64 | 4.4e-07 | 4.4e-07 | 1.2e-12 | 1.2e-12 | 7.9e-15 | 7.9e-15 |

$g_1$ and $g_2$ of (4.1) and (4.2) on the CONT-201 matrix. Thus the left-hand figures show the real growth factors, while the pivoting strategy may take its decision according to a different estimation. With the `numSEQ` strategy the estimation and the actual values are exactly the same. That is why we always have the actual growth factors smaller than $10^2$. With `numBPAR` and `numEPAR`, the actual value may be underestimated. That is why we observe growth factors greater than $10^2$. Figure 6.2 confirms that the `numSEQ` and `numEPAR` strategies have similar behavior in the sense that they postpone approximately the same number of pivots at each node. Furthermore the `numEPAR` strategy succeeds in bounding the growth factor by $4 \times 10^3$ at each step of Gaussian elimination on CONT-201.

The behavior of the `numBPAR` strategy is significantly different. We see, in Table 6.2, that the number of delayed pivots decreases significantly on the BRATU3D matrix. Consequences of this are a decrease in the number of nonzeros in the factors ($9.7 \times 10^6$ for `numBPAR` versus $11.5 \times 10^6$ for `numSEQ`) and a degradation of the precision, as seen in Table 6.1.

For the CONT-201 and CONT-300 matrices, `numBPAR` delays approximately the same number of variables and computes factors with a similar number of nonzeros. Figure 6.2 shows that `numBPAR` does not guarantee a reasonable growth factor on CONT-201 (growth factors nearly equal to $1/\epsilon$ for some nodes of the assembly tree). We see also that the number of delayed pivots is significantly different between the `numBPAR` strategy and the two other strategies, `numSEQ` and `numEPAR`, for some nodes of the tree.

**6.3.2. Parallel approaches combining numerical pivoting and perturbation techniques.** Table 6.3 shows that there are no significant differences between the sequential and the parallel versions of our Algorithm 1. We still need two iterations to converge to the machine precision on most of the matrices.

**7. Influence of preprocessing.** In [12] we presented preprocessing techniques to improve the quality of preselected pivots. Our preprocessing uses symmetric weighted matching and sparsity ordering techniques. Of the techniques presented, we saw that METIS combined with the `MC64SYM` scaling was the best ordering on symmetric indefinite matrices in terms of CPU factorization time, but that it sometimes caused many pivots to be delayed. We also proposed an ordering based on METIS that increases the number of nonzeros in the factors slightly but that clearly improves the

Table 7.1

*Influence of the preprocessing on the component-wise backward error and on the number of perturbed pivots.* MeTiS*:* `mixSEQ` *with* MeTiS *ordering.* `CMP`*:* `mixSEQ` *with a preprocessing based on symmetric weighted matching and on* MeTiS*.*

| | Iteration 0 | | Iteration 1 | | Iteration 2 | Tiny pivots | |
|---|---|---|---|---|---|---|---|
| Matrix | MeTiS | `CMP` | MeTiS | `CMP` | MeTiS | MeTiS | `CMP` |
| BRAINPC2 | 2.9e-06 | 1.7e-11 | 3.2e-12 | 1.7e-15 | 2.0e-15 | 12932 | 0 |
| BRATU3D | 1.6e-05 | 1.2e-07 | 6.9e-12 | 1.2e-15 | 2.9e-16 | 8429 | 284 |
| CONT-201 | 2.7e-05 | 8.3e-14 | 4.0e-08 | 2.2e-16 | 3.2e-08 | 27470 | 0 |
| CONT-300 | 6.4e-05 | 3.5e-13 | 8.6e-08 | 2.5e-16 | 8.6e-08 | 67864 | 0 |
| cvxqp3 | 1.3e-03 | 1.1e-04 | 4.4e-12 | 2.2e-13 | 4.4e-16 | 6277 | 30 |
| mario001 | 1.6e-07 | 1.4e-07 | 4.8e-12 | 1.0e-14 | 3.3e-16 | 10305 | 29 |
| NCVXQP1 | 9.9e-01 | 9.8e-01 | 1.1e-04 | 3.9e-06 | 1.8e-11 | 3619 | 10 |
| NCVXQP5 | 5.7e-05 | 2.9e-10 | 1.6e-08 | 3.2e-16 | 5.3e-12 | 8402 | 0 |
| NCVXQP7 | 2.0e-02 | 2.8e-06 | 1.5e-10 | 6.5e-14 | 8.4e-16 | 31043 | 46 |
| SIT100 | 3.0e-07 | 8.5e-07 | 6.7e-14 | 5.2e-14 | 2.1e-16 | 1388 | 6 |
| stokes128 | 6.0e-07 | 1.8e-07 | 3.0e-12 | 9.4e-13 | 4.0e-15 | 12738 | 27 |
| stokes64 | 4.4e-07 | 2.1e-07 | 1.2e-12 | 5.1e-13 | 7.9e-15 | 3106 | 5 |

Table 7.2

*Influence of the preprocessing on the factorization, the solution time (in seconds), and the memory required by the factorization.* MeTiS: `mixSEQ` *with* MeTiS *ordering.* `CMP`: `mixSEQ` *with a preprocessing based on symmetric weighted matching and on* MeTiS*. (x) indicates that the method requires x steps of iterative refinement to get a componentwise backward error smaller than* $10^{-10}$*. (-) indicates that this level of precision is not reached.*

| | Factorization time | | Time for forward and backward substitution | | Memory required in MBytes | |
|---|---|---|---|---|---|---|
| Matrix | MeTiS | `CMP` | MeTiS | `CMP` | MeTiS | `CMP` |
| BRAINPC2 | 0.11 | 0.11 | 0.014 (1) | 0.015 (0) | 5.2 | 5.2 |
| BRATU3D | 9.24 | 8.56 | 0.125 (1) | 0.124 (1) | 35.6 | 35.3 |
| CONT-201 | 1.94 | 1.44 | 0.127 (-) | 0.121 (0) | 27.9 | 27.3 |
| CONT-300 | 6.08 | 4.38 | 0.306 (-) | 0.294 (0) | 67.3 | 65.6 |
| cvxqp3 | 3.08 | 9.46 | 0.048 (1) | 0.110 (1) | 17.3 | 32.5 |
| mario001 | 0.23 | 0.31 | 0.022 (1) | 0.029 (1) | 6.8 | 8.6 |
| NCVXQP1 | 1.29 | 5.44 | 0.024 (2) | 0.055 (2) | 10.6 | 20.7 |
| NCVXQP5 | 23.0 | 39.8 | 0.279 (2) | 0.425 (0) | 68.2 | 102.1 |
| NCVXQP7 | 71.6 | 199. | 0.498 (1) | 0.988 (1) | 124.9 | 250.3 |
| SIT100 | 0.11 | 0.13 | 0.004 (1) | 0.005 (1) | 3.0 | 3.4 |
| stokes128 | 1.06 | 1.63 | 0.076 (1) | 0.111 (1) | 20.4 | 27.4 |
| stokes64 | 0.29 | 0.38 | 0.013 (1) | 0.017 (1) | 4.7 | 6.1 |

numerical stability for the preselected pivots. In this section, we study the influence of this preprocessing in the context of our new pivoting strategies. We use these two orderings for the present experiments: the MeTiS ordering and an ordering that we call MeTiS on the compressed graph (denoted by `CMP`; see [12] for more details).

Table 7.1 compares our restricted pivoting when using the above two orderings. We see that the preselection of $2 \times 2$ pivots using a symmetric weighted matching significantly decreases the number of tiny pivots and improves the precision of the solution. This influence of pivot preselection using maximum weighted matching techniques has also been observed in the context of `SuperLU_DIST` [22]. We see that, in most of the cases, the approach with an ordering on the compressed graph needs one iteration fewer than an approach based on the MeTiS ordering.

Generally the approach on the compressed graph increases the number of operations, the fill-in in the factors, and the memory needed. It also increases the factorization time (see Table 7.2). Note that the compressed approach is better on

CONT-201 and CONT-300 in terms of fill-in in the factors and factorization time, because the compression detects cliques and then improves the quality of the fill-in reducing phase (METIS).

We also compare the solution times in Table 7.2. Generally the `CMP` strategy increases the size of the factors and thus increases the time for backward and forward substitution. Nevertheless this effect is compensated by the decrease in the number of iterative refinement steps: on average, the `CMP` strategy requires one backward and forward substitution and one matrix-vector multiplication fewer than the METIS strategy to converge to a certain level of precision.

We also studied the influence of the preprocessing on parallel approaches. As in the sequential case, our preprocessing improves the numerical robustness, and we get small backward errors after one or two iterative refinement steps.

**8. Conclusions.** We have presented different pivoting strategies that combine traditional sparse pivoting strategies and perturbation techniques and have identified their main characteristics. We have implemented these within the `MA57` code. Our new pivoting strategies seem to address a large class of challenging problems and to be significantly faster than approaches with standard numerical pivoting.

We recommend checking the backward error of the computed solution. If it is too large, then one or two iterative refinement steps are enough to converge to an accurate solution. Note also that the target may not be the machine precision. For example, if we consider Newton iterations, an approximate solution of the linear system may be sufficient at the beginning. For classes of problems on which the `mixEPAR` or `mixSEQ` strategies do not compute a precise enough solution, we have proposed alternatives:

- preprocessing the matrix using a compressed graph and a maximum weighted matching,
- and/or using the numerical pivoting strategies, `numEPAR` or `numSEQ`.

Our parallel approaches can easily be generalized to the unsymmetric case. Restricted pivoting within `SuperLU_DIST` would improve its precision. Concerning the unsymmetric version of `MUMPS` it would also allow 2-dimensional partitioning of type 2 nodes (in particular, a 2-dimensional partitioning of the master task) and improve the scalability of `MUMPS`.

Even if the precision of the solution is good, further improvements can be obtained. First, there are still problems on which restricted pivoting is less accurate than approaches that perform standard threshold pivoting. Second, the number of iterative refinement steps can be decreased on some problems. This problem becomes all the more critical when we have many right-hand sides. As a much higher value for $\mu$ would result in a far more stable factorization but perhaps with a modified matrix further from the original, we intend to study the trade-off whose optimal value might well be quite different if iterative methods more sophisticated than iterative refinement (for example, MINRES, GMRES) were used.

**Acknowledgments.** We would like to thank John Reid and Jennifer Scott of the Rutherford Appleton Laboratory for their comments on a draft version of this manuscript. We are also very grateful to Serge Gratton for discussions on various aspects of this work, and to the anonymous referees for their helpful comments.

## REFERENCES

[1] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.

[2] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 501–520.

[3] M. Arioli, J. W. Demmel, and I. S. Duff, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.

[4] C. Ashcraft, R. G. Grimes, and J. G. Lewis, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 513–561.

[5] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comput., 31 (1977), pp. 162–179.

[6] J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[7] S. H. Cheng and N. J. Higham, *A modified Cholesky algorithm based on a symmetric indefinite factorization*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 1097–1110.

[8] T. A. Davis, *University of Florida Sparse Matrix Collection*, 2002, http://www.cise.ufl.edu/research/sparse/matrices.

[9] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Numerical Linear Algebra on High-Performance Computers*, Software Environ. Tools 7, SIAM, Philadelphia, 1998.

[10] I. S. Duff, *MA57—A code for the solution of sparse symmetric indefinite systems*, ACM Trans. Math. Software, 30 (2004), pp. 118–144.

[11] I. S. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.

[12] I. S. Duff and S. Pralet, *Strategies for scaling and pivoting for sparse symmetric indefinite problems*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 313–340.

[13] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[14] I. S. Duff and J. K. Reid, *Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Software, 22 (1996), pp. 227–257.

[15] E. Eskow and R. B. Schnabel, *Algorithm 695: Software for a new modified Cholesky factorization*, ACM Trans. Math. Software, 17 (1991), pp. 306–312.

[16] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 28 (1974), pp. 311–350.

[17] N. I. M. Gould, J. A. Scott, and Y. Hu, *A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations*, ACM Trans. Math. Softw., 33 (2007).

[18] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr (and SifDec), A Constrained and Unconstrained Testing Environment, Revisited*, Technical Report 2002-009, Rutherford Appleton Laboratory, Oxon, UK, 2002.

[19] N. I. M. Gould and J. A. Scott, *A numerical evaluation of HSL packages for the direct solution of large sparse, symmetric linear systems of equations*, ACM Trans. Math. Software, 30 (2004), pp. 300–325.

[20] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[21] X. S. Li and J. W. Demmel, *Making sparse Gaussian elimination scalable by static pivoting*, in Proceedings of Supercomputing, San Jose, CA, 1998, IEEE Computer Society Press, Washington, DC, 1998, pp. 1–17.

[22] X. S. Li and J. W. Demmel, *SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems*, ACM Trans. Math. Software, 29 (2003).

[23] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[24] J. W. H. Liu, E. G. Ng, and B. W. Peyton, *On finding supernodes for sparse matrix computations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 242–252.

[25] I. Maros and C. Meszaros, *A repository of convex quadratic programming problems*, Optim. Methods Softw., 11–12 (1999), pp. 671–681.

[26] S. Pralet, *Constrained Orderings and Scheduling for Parallel Sparse Linear Algebra*, Ph.D thesis, Institut National Polytechnique de Toulouse, Toulouse, France, 2004; also available as CERFACS Technical Report TH/PA/04/105.

[27] O. Schenk and K. Gärtner, *On Fast Factorization Pivoting Methods for Sparse Symmetric Indefinite Systems*, Technical Report CS-2004-004, Computer Science Department, University of Basel, Basel, Switzerland, 2004.

[28] G. W. Stewart, *Modifying pivot elements in Gaussian elimination*, Math. Comput., 28 (1974), pp. 537–542.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 3, pp. 1025–1049

# ON THE CONVERGENCE OF A CLASS OF MULTILEVEL METHODS FOR LARGE SPARSE MARKOV CHAINS*

PETER BUCHHOLZ† AND TUĞRUL DAYAR‡

**Abstract.** This paper investigates the theory behind the steady state analysis of large sparse Markov chains with a recently proposed class of multilevel methods using concepts from algebraic multigrid and iterative aggregation-disaggregation. The motivation is to better understand the convergence characteristics of the class of multilevel methods and to have a clearer formulation that will aid their implementation. In doing this, restriction (or aggregation) and prolongation (or disaggregation) operators of multigrid are used, and the Kronecker-based approach for hierarchical Markovian models is employed, since it suggests a natural and compact definition of grids (or levels). However, the formalism used to describe the class of multilevel methods for large sparse Markov chains has no influence on the theoretical results derived.

**Key words.** Markov chains, multigrid, aggregation-disaggregation, Kronecker-based numerical techniques, multilevel methods

**AMS subject classifications.** 60J27, 65F50, 65F10, 65B99, 65F15, 65F05, 15A72

**DOI.** 10.1137/060651161

## 1. Introduction.

Markov chains (MCs) are a popular mathematical tool to model systems from various application areas like engineering, computer science, biology, or economics. For system analysis often one needs the steady state distribution of the MC to compute result measures for the modeled system. The problem in the continuous-time case is then to solve

$$\pi Q = 0 \quad \text{subject to} \quad \pi e = 1 \quad \text{and} \quad \pi \geq 0, \tag{1.1}$$

where $Q$ is the infinitesimal generator or generator matrix of the continuous-time Markov chain (CTMC) underlying the modeled system, $\pi$ is its (row) stationary probability vector, and $e$ is the column vector of ones of appropriate length. We assume that the state space is finite and contains $n$ states numbered starting from 0; $Q$ is irreducible, implying $\pi > 0$; and $\pi$ is also the steady state vector. The nonnegative off-diagonal elements of $Q$ represent exponential transition rates between different states, and its diagonal elements are negated row sums of its off-diagonal elements. Hence, $Q$ has row sums of zero (i.e., $Qe = 0$) and is a singular matrix of rank $(n-1)$, and (1.1) represents a homogeneous linear system subject to a normalization condition, so that its solution vector $\pi$ can be uniquely determined [29, Chap. 1]. At this level, states of the CTMC are numbered by consecutive integers. However, in almost all applications CTMCs result from some high level model like a stochastic automata network, a queueing network, or a stochastic Petri net. In all these cases, the state space is multidimensional and is mapped for solution onto a set of consecutive

---

*Received by the editors January 30, 2006; accepted for publication (in revised form) by D. A. Bini March 13, 2007; published electronically October 31, 2007. Part of this work has been carried out through a grant from the Alexander von Humboldt Foundation at Dortmund University and grant TÜBA-GEBİP from the Turkish Academy of Sciences.

http://www.siam.org/journals/simax/29-3/65116.html

†Informatik IV, Universität Dortmund, D-44221 Dortmund, Germany (peter.bucholz@cs.uni-dortmund.de).

‡Department of Computer Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey (tugrul@cs.bilkent.edu.tr.).

integers. The multidimensional structure can be exploited in a compact representation of $Q$ and can also be exploited to develop fast solvers for the computation of $\pi$.

Practical problems arise due to the state space size of MCs resulting from applications, which often grows exponentially with the number of components in the specification. A popular way of dealing with this so-called state space explosion problem is to employ Kronecker- (or tensor)-based representations of $Q$, which remain compact even for considerably large state spaces. In the Kronecker-based approach, the system of interest is modeled so that it is formed of smaller interacting components, and its larger underlying generator matrix is neither generated nor stored but rather represented using Kronecker products of the smaller component matrices. This introduces significant storage savings at the expense of some overhead in the solution phase. In order to analyze large structured Markovian models efficiently, various algorithms for vector-Kronecker product multiplication are devised [14, 16, 17] and used as kernels in iterative solution methods. The most effective solvers known for Kronecker representations of dimension four or larger are multilevel (ML) methods [11] and block successive over relaxation (BSOR) preconditioned projection methods [12] as recently shown empirically by comparing different solvers on a large number of hierarchical Markovian models (HMMs). Unfortunately, solvers using BSOR [10, 31] are sensitive to the ordering of components, the block partitionings chosen, and the amount of fill-in in the factorized diagonal blocks, so that a robust implementation for arbitrary models is difficult to achieve.

In this paper, we investigate the theory behind the steady state analysis of large sparse MCs with the class of ML methods proposed in [11] using concepts from algebraic multigrid (AMG) [6, Chap. 8], [24] and iterative aggregation-disaggregation (IAD) [29, Chap. 6]. Our motivation is to better understand the convergence characteristics of the class of ML methods and to have a clearer formulation that will aid their implementation. Convergence analysis of a two-level IAD method for MCs and its equivalence to AMG is provided in [20]. Another paper that investigates the convergence of a two-level IAD method for MCs using concepts from multigrid is [21]. Here we consider more than two levels, different types of smoothers, different types of cycles, and different orders of aggregation. In doing this, we use restriction (or aggregation) and prolongation (or disaggregation) operators of multigrid, and employ the Kronecker-based approach for HMMs in [11]. This is for three reasons. First, the hierarchy present in the HMM description suggests a natural definition of grids (or levels). This simplifies the description of the class of ML methods. Second, with the HMM description, one can store the aggregated MC at each level during implementation compactly in Kronecker form. It is not clear how the same effect can be achieved with an MC in sparse format (see [19]). Third, Kronecker operations to define large MCs underlying structured representations are natural for many application areas since complex systems are usually composed of interacting components. Almost all MCs resulting from applications can be represented as HMMs [15], and this representation can be derived from the specification using an appropriate modeling tool [1]. Otherwise, the HMM formalism used in this paper to describe the class of ML methods for large sparse MCs has no influence on the theoretical results derived. In general, our approach can be applied for any irreducible MC with a set of nested partitions defined on its state space.

The next section introduces the Kronecker-based description of CTMCs underlying HMMs on a small example. The third section presents the proposed class of ML methods for HMMs with multiple macrostates and discusses how they work. The fourth section provides results on the convergence of ML methods. The fifth section illustrates the convergence behavior of the class of ML methods on two larger problems. The sixth section concludes the paper.

In what follows, calligraphic uppercase letters denote sets and lists, uppercase letters denote matrices, sets are defined using curly brackets, lists are defined using square brackets, matrices (and vectors) are defined using brackets, $|\cdot|$ denotes the cardinality of a set (list) when its argument is a set (list), $\emptyset$ denotes the empty set, $||\cdot||$ denotes the norm of a vector, $\cdot^T$ denotes the transpose operator, and diag$(\cdot)$ represents a diagonal matrix having its vector argument along its diagonal.

**2. Hierarchical Markovian models.** HMMs are defined using the operations of Kronecker product and Kronecker sum [32]. First we introduce these operations.

Definition 2.1. [illegible] $X \in \mathbb{R}^{r_X \times c_X}$ [illegible] $Y \in \mathbb{R}^{r_Y \times c_Y}$ [illegible] $X \otimes Y$ [illegible] $Z$ [illegible] $r_X \times c_X$ [illegible] $r_Y \times c_Y$ [illegible] $(i,j)$ [illegible] $x(i,j)Y$ [illegible] $i = 0,\ldots,r_X-1$ $j = 0,\ldots,c_X-1$ [illegible] $U \in \mathbb{R}^{r_U \times r_U}$ [illegible] $V \in \mathbb{R}^{r_V \times r_V}$ [illegible] $U \oplus V$ [illegible] $W \in \mathbb{R}^{r_U r_V \times r_U r_V}$ [illegible] $W = U \otimes I_{r_V} + I_{r_U} \otimes V$ [illegible] $I_{r_U}$ [illegible] $I_{r_V}$ [illegible] $r_U$ [illegible] $r_V$ [illegible]

Both Kronecker product and Kronecker sum are associative and defined for more than two matrices. For further properties of Kronecker operations, see [29].

HMMs consist of multiple low level models (LLMs) which can be perceived as components, and a high level model (HLM) that defines how LLMs interact. The HLM is characterized by a single matrix, whereas each LLM is characterized by multiple matrices that define its interaction with other LLMs. The order of each LLM matrix is equal to the number of states of the particular component to which the matrix belongs. A formal definition of HMMs can be found in [8, pp. 387–390]. Here we extend the definition from [12] and introduce HMMs on an example. An HMM describes a CTMC and its generator matrix $Q$. Since we consider the steady state analysis of irreducible finite CTMCs, $Q$ is sufficient to characterize the CTMC. We name the states of the HLM as [illegible], those of $Q$ as [illegible], and remark that macrostates define a partition of the microstates.

Definition 2.2. [illegible] $K$ [illegible] $\mathcal{S}^{(k)} = \{0,1,\ldots,|\mathcal{S}^{(k)}|-1\}$ [illegible] $k$ [illegible] $k = 1,2\ldots,K$ $\mathcal{S}^{(K+1)} = \{0,1,\ldots,|\mathcal{S}^{(K+1)}|-1\}$ [illegible] $\mathcal{S}_j^{(k)}$ [illegible] $k$ [illegible] $j \in \mathcal{S}^{(K+1)}$ [illegible] $\cup_j \mathcal{S}_j^{(k)} = \mathcal{S}^{(k)}$ [illegible] $\mathcal{S}_i^{(k)} \cap \mathcal{S}_j^{(k)} = \emptyset$ [illegible] $i \neq j$ $t_0$ [illegible] $\mathcal{T}_{i,j}$ [illegible] $(i,j)$ [illegible] $D_j$ [illegible] $Q$ [illegible] $j$ [illegible] $(j,j)$ [illegible] $Q$ [illegible] $(j,j)$ [illegible]

$$Q(j,j) = \bigoplus_{k=1}^{K} Q_{t_0}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_j^{(k)}) + \sum_{t_e \in \mathcal{T}_{j,j}} \bigotimes_{k=1}^{K} Q_{t_e}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_j^{(k)}) + D_j, \tag{2.1}$$

[illegible] $(i,j)$ [illegible] $Q$ [illegible] $(i,j)$ [illegible]

$$Q(i,j) = \sum_{t_e \in \mathcal{T}_{i,j}} \bigotimes_{k=1}^{K} Q_{t_e}^{(k)}(\mathcal{S}_i^{(k)}, \mathcal{S}_j^{(k)}), \tag{2.2}$$

[illegible] $Q_{t_e}^{(k)}(\mathcal{S}_i^{(k)}, \mathcal{S}_j^{(k)})$ [illegible] $(|\mathcal{S}_i^{(k)}| \times |\mathcal{S}_j^{(k)}|)$ [illegible][1] [illegible] $\mathcal{S}_i^{(k)}$ [illegible] $\mathcal{S}_j^{(k)}$ [illegible] $k$ [illegible] $t_e$

We remark that $D_j$ can be expressed as a sum of Kronecker products, as follows.

PROPOSITION 2.3. [illegible] $D_j$ [illegible] $Q$ [illegible] $j$ [illegible]

$$D_j = -\bigoplus_{k=1}^{K} \mathrm{diag}(Q_{t_0}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_j^{(k)})e)$$

$$- \sum_{i \in \mathcal{S}^{(K+1)}} \sum_{t_e \in \mathcal{T}_{j,i}} \bigotimes_{k=1}^{K} \mathrm{diag}(Q_{t_e}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_i^{(k)})e) \quad \text{[illegible]} \; j \in \mathcal{S}^{(K+1)}.$$

In order to enable the efficient implementation of numerical solvers, most of the time $D_j$ is precomputed and stored explicitly as a vector. However, the off-diagonal part of $Q$ is never stored explicitly, but is represented in memory through Definition 2.2 as sums of Kronecker products of small matrices, which are generally very sparse and therefore held in row sparse format [29, pp. 80–81].

For a definition of mapping used in the next proposition, see, for instance, [30, pp. 192–197].

PROPOSITION 2.4. [illegible] $Q$ [illegible] $(s^{(1)}, s^{(2)}, \ldots, s^{(K)}, j)$ [illegible] $s^{(k)} \in \mathcal{S}^{(k)}$ [illegible] $k$ [illegible] $k = 1, 2, \ldots, K$ [illegible] $j \in \mathcal{S}^{(K+1)}$ [illegible] $Q$ [illegible]

$$(s^{(1)}, s^{(2)}, \ldots, s^{(K)}, j) \longleftrightarrow \sum_{k=1}^{K} s^{(k)} \prod_{l=k+1}^{K} |\mathcal{S}_j^{(l)}| + \sum_{i=0}^{j-1} \prod_{k=1}^{K} |\mathcal{S}_i^{(k)}| \in \{0, 1, \ldots, n-1\},$$

[illegible] $n = \sum_{j=0}^{|\mathcal{S}^{(K+1)}|-1} \prod_{k=1}^{K} |\mathcal{S}_j^{(k)}|$.

The microstates corresponding to each macrostate result from the Cartesian (or cross) product [30, pp. 123–124] of the state space partitions of LLMs that are mapped to that particular macrostate. In contrast to other representations of CTMCs using Kronecker operators (e.g., [29, Chap. 9]), HMMs are generated in such a way that only reachable states are considered [7, 8]. Note that each macrostate in an HLM may have a different number of microstates if LLMs have partitioned state spaces. When there are multiple macrostates, $Q$ is effectively a block matrix having as many blocks in each dimension as $|\mathcal{S}^{(K+1)}|$. The diagonal and off-diagonal blocks of this partitioning are respectively the $Q_{j,j}$ and $Q_{i,j}$ matrices defined by (2.1) and (2.2). Due to the Kronecker structure suggested by Definitions 2.1 and 2.2, each of the blocks defined by the HLM matrix is also formed of blocks, and hence HMMs have nested block partitionings [10, 31].

Now, let us consider a small example HMM which gives rise to a $(5 \times 5)$ CTMC. In [13, sec. 5], we step through the ML method on this example, which is chosen deliberately to be very small. After this small example, we briefly present two larger examples which will be used in section 5 to show the convergence behavior of the class of ML methods.

---

[1] In this section, the concept of transition is used to refer to those that take place at the HMM level, except for this case, where it is used to refer to nonzeros in a matrix at the state level.

Table 2.1
*Mapping between LLM states and HLM states in Example* 1.

| LLM 1 | LLM 2 | HLM | # of microstates |
|---|---|---|---|
| {0,1} | {0,1} | {0} | 2 . 2 = 4 |
| {2} | {2} | {1} | 1 . 1 = 1 |

[illegible] 1. The HLM of two states describes the interaction among two LLMs (i.e., $K = 2$), each of which has three states. All states are numbered starting from 0. The mapping between LLM states and HLM states and the number of microstates are given in Table 2.1. In this example, $Q$ has the following states in its rows and columns: $\{0,1\} \times \{0,1\} \times \{0\} \cup \{2\} \times \{2\} \times \{1\} = \{(0,0,0),(0,1,0),(1,0,0),(1,1,0),(2,2,1)\}$. One can think of these five states written in the given order as corresponding to the integers 0 through 4.

The values of the nonzeros in $Q$ are determined by the rates of the transitions and their associated matrices. In Example 1, two transitions denoted by $t_0$ and $t_1$ take place and affect the LLMs. Transition $t_0$ covers all local transitions inside the LLMs, whereas transition $t_1$ is captured by the following $(2 \times 2)$ HLM matrix:

$$\begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \left(\begin{array}{c|c} & t_1 \\ \hline t_1 & \end{array}\right) \end{array}. \tag{2.3}$$

To each transition in the HLM matrix corresponds a Kronecker product of two (i.e., number of LLMs, $K$) LLM matrices. The matrices associated with those LLMs that do not participate in a transition are identity. LLM 1 participates in $t_1$ with the matrix $Q_{t_1}^{(1)}$, and LLM 2 participates in $t_1$ with the matrix $Q_{t_1}^{(2)}$. In this example, the transition $t_1$ affects exactly two LLMs.

Other than Kronecker products due to the transitions in (2.3), there is a Kronecker sum implicitly associated with each diagonal element of the HLM matrix. Each Kronecker sum is formed of two (i.e., $K$) LLM matrices corresponding to [illegible] $t_0$. In the HLM matrix of (2.3), there does not exist any nonlocal transition along the diagonal. In general, this need not be so, as can be seen from Definition 2.2.

In our example, the second term in (2.1) is missing, and the matrices associated with $t_0$ and $t_1$ are given by

$$Q_{t_0}^{(1)} = \left(\begin{array}{cc|c} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right), \quad Q_{t_1}^{(1)} = \left(\begin{array}{cc|c} 0 & 0 & 2 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \end{array}\right), \quad Q_{t_0}^{(2)} = \left(\begin{array}{cc|c} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \end{array}\right),$$

$$Q_{t_1}^{(2)} = \left(\begin{array}{cc|c} 0 & 0 & 1 \\ 0 & 0 & 0 \\ \hline 1 & 0 & 0 \end{array}\right).$$

Then the CTMC underlying the HMM can be obtained from

$$Q = \left(\begin{array}{c|c} Q_{t_0}^{(1)}(\{0,1\},\{0,1\}) \bigoplus Q_{t_0}^{(2)}(\{0,1\},\{0,1\}) & Q_{t_1}^{(1)}(\{0,1\},\{2\}) \bigotimes Q_{t_1}^{(2)}(\{0,1\},\{2\}) \\ \hline Q_{t_1}^{(1)}(\{2\},\{0,1\}) \bigotimes Q_{t_1}^{(2)}(\{2\},\{0,1\}) & Q_{t_0}^{(1)}(\{2\},\{2\}) \bigoplus Q_{t_0}^{(2)}(\{2\},\{2\}) \end{array}\right) + D, \tag{2.4}$$

where $D$ is the diagonal correction matrix that sums the rows of $Q$ to zero; hence, written explicitly, we have

$$Q = \left(\begin{array}{rrrr|r} -4 & 1 & 1 & 0 & 2 \\ 1 & -2 & 0 & 1 & 0 \\ 1 & 0 & -3 & 1 & 1 \\ 0 & 1 & 1 & -2 & 0 \\ \hline 1 & 0 & 0 & 0 & -1 \end{array}\right). \tag{2.5}$$

If we neglect the diagonal of $Q$, which is handled separately, from Definition 2.2 it follows that each nonzero element of the HLM matrix is essentially a sum of Kronecker products, since Kronecker sums can be expressed as sums of Kronecker products. This has a very nice implication for the choice of grids in the proposed ML method when LLM aggregation is used in forming the coarse grids. LLMs 1 through $K$ and the HLM define the least coarse (in other words, the finest) grid. This grid is $Q$ and in our example has five states. Regarding the intermediate grids, let us assume that LLMs are aggregated starting from 1 up to $K$. Thus LLMs 2 through $K$ and the HLM define the first coarser grid when LLM 1 is aggregated. In our example, this grid has the states in $\{(0,0),(1,0),(2,1)\}$, where the first state in each tuple is an LLM 2 state and the second state in each tuple is the corresponding HLM state. The HLM and LLMs 3 through $K$ define the second coarser grid when LLMs 1 and 2 are aggregated. In our example, this grid is the coarsest grid corresponding to the HLM and has the states $\{(0),(1)\}$. There are no other LLMs left to be aggregated in our example; otherwise aggregation continues with the next LLM.

Now, let us concentrate on the sizes of the grids defined by the LLMs and the HLM for the assumed order in which LLMs are aggregated. In Example 1, the grids defined in this way by LLMs 1–2 and the HLM, by LLM 2 and the HLM, and by the HLM alone have respectively the sizes $(5\times 5)$, $(3\times 3)$, $(2\times 2)$ (see Table 2.1 and (2.1)–(2.2)). Clearly, we are not limited to aggregating LLMs in the order 1 through $K$, and can consider other orderings. The number of possible orderings of LLMs equals $K!$.

*Example* 2. The second example we consider is a polling system. Two servers serve customers from $K$ finite capacity queues, which are visited by the servers in cyclic order. We assume that each queue has a capacity of 3, and customers arrive according to a Poisson process with rate 1.5 and are distributed with queue specific probabilities among the queues. If a server visits a nonempty queue, it serves one customer and then moves to the next queue. A server arriving at an empty queue immediately travels to the next queue. Service and traveling times are exponentially distributed with rates 1 and 10, respectively. Each LLM describes one queue, and macrostates for this model are defined according to the number of servers serving customers at a queue or traveling to the next queue. For each LLM we obtain 20 states partitioned into three subsets. The complete model has $\binom{K+1}{K-1}$ macrostates. Table 2.2 contains the number of microstates for different values of $K$.

*Example* 3. The third example describes an availability model with $K$ LLMs. Each LLM consists of two active components and a cold spare which becomes active when a component fails. Time to failure is exponentially distributed with mean $10^k$ for the components of the $k$th LLM. With 90% probability a failure is local, requiring a local repair with an exponential duration and mean $10^{-k+1}$ for the $k$th component. With a probability of 10%, a failure has to be repaired by a global repair unit; repair times are identical to the local case. The system has one global repair unit which repairs failed components with preemptive priority such that components from the

Table 2.2
*Number of macrostates and state space sizes versus number of LLMs in Examples* 2 *and* 3.

| $K$ | Polling example $\lvert\mathcal{S}^{(K+1)}\rvert$ | Polling example $n$ | Availability example $\lvert\mathcal{S}^{(K+1)}\rvert$ | Availability example $n$ |
|---|---|---|---|---|
| 2 | – | – | 1 | 100 |
| 3 | 6 | 1,020 | 1 | 1,000 |
| 4 | 10 | 7,008 | 1 | 10,000 |
| 5 | 15 | 42,880 | 1 | 100,000 |
| 6 | 21 | 243,456 | 1 | 1,000,000 |
| 7 | 28 | 1,311,744 | 1 | 10,000,000 |

first LLM get the highest priority and components from the $K$th LLM obtain the least priority. As can be seen in Table 2.2, the system contains one macrostate and $10^K$ microstates. Note that this is an example in which different time scales occur and is therefore expected to be harder to solve by classical iterative methods.

In the next section, we introduce the class of ML methods with the grid choices suggested by the Kronecker structure of HMMs and remark that, just like $Q$, none of the grids except the coarsest is explicitly generated.

## 3. A class of ML methods.

The class of ML methods presented in this section are related to IAD for the analysis of MCs [29, sec. 6.3] and AMG for general systems of equations [24]. IAD is applied in the context of MCs to coefficient matrices with a two-level block structure, where blocks are loosely coupled. Different variants of the method exist; all combine the solution of an aggregated system, whose elements correspond to blocks in the two-level block partitioning, with iteration steps or solutions of systems of equations at the block level. The solution of the aggregated system distributes the steady state probability mass over the loosely coupled subsets of states, whereas at the block level the probability mass is distributed inside the subsets. AMG solves a system of equations by performing iterations on systems of equations of decreasing size. Our approach can be interpreted as a specific form of AMG for singular M-matrices, a class of matrices which will be defined in the next section. However, like in geometric multigrid, our grids have a physical meaning, since they are defined according to subsets of LLMs. Furthermore, the grids may change from one ML iteration to the next by varying the order in which LLMs are aggregated. Like in geometric multigrid, the goal is to achieve convergence that is independent of the size of the original problem. This means that the number of ML iterations to reach a predefined tolerance should be more or less independent of the number of LLMs for a given model structure. The proposed class of ML methods are related to IAD, since aggregation-disaggregation steps are used to realize the mapping between different levels. However, in contrast to IAD, varying and possibly more than two levels are defined, and the Kronecker structure is exploited to represent the aggregated matrix at each level. This implies that the class of ML methods are also expected to be efficient for large models where LLMs are tightly coupled.

### 3.1. Algorithms.

One iteration of AMG over a system of linear equations is referred to as a [illegible]. Throughout the text, we use [illegible] and [illegible] interchangeably. The order in which the smaller aggregated systems are visited during each AMG iteration gives rise to different cycle types. Within an AMG cycle, the iterative method used to improve the solution of each aggregated system is called a [illegible], since it is perceived to smooth the error in the solution at that level. The class of ML methods for HMMs with multiple macrostates have the capability of using (V, W, F) cycles

[33], (power, Jacobi over relaxation (JOR), successive over relaxation (SOR)) methods as smoothers, and (fixed, circular, dynamic) orders in which LLMs can be aggregated in an ML iteration. These parameters are respectively denoted by $C$ for cycle type, $S$ for smoother type, and $O$ for order of aggregating LLMs. Hence, $C \in \{V, W, F\}$, $S \in \{POWER, JOR, SOR\}$, and $O \in \{FIXED, CIRCULAR, DYNAMIC\}$. In a particular ML solver, $C$, $S$, and $O$ are fixed at the beginning and do not change.

Algorithm 1 is the driver of the ML solver. It starts executing at the finest grid involving the LLMs and the HLM, and then invokes the recursive ML function in Algorithm 2 with the order in which LLMs are to be aggregated in the list $\mathcal{C}$. Each pass through the body of the repeat-until loop in Algorithm 1 corresponds to one ML iteration (i.e., cycle). Observe that steps 3 through 8 in Algorithm 2 are almost identical to the statements between steps 3 and 4 in Algorithm 1.

Algorithm 1. [illegible]

```
main()
$\mathcal{D} = [1, 2, \ldots, K+1]$; $\tilde{Q}_{\mathcal{D}} = Q$; $x_{\mathcal{D}}$ = initial approximation; $it = 0$; $stop = FALSE$;
(step 1)
if ($C == W$ or $C == F$) then                                          (step 2)
   $\gamma = 2$;
else
   $\gamma = 1$;
repeat                                                                  (step 3)
   $x'_{\mathcal{D}} = S(\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}}, w, \nu_1)$;
   remove $\mathcal{D}_1$ from $\mathcal{D}$ by aggregation to give $\mathcal{C}$;
   $\tilde{Q}_{\mathcal{C}} = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} R_{\mathcal{D}}$; $x_{\mathcal{C}} = x'_{\mathcal{D}} R_{\mathcal{D}}$;
   if ($\gamma == 1$) then
      $y_{\mathcal{C}} = \mathrm{ML}(\tilde{Q}_{\mathcal{C}}, x_{\mathcal{C}}, \mathcal{C}, \gamma)$;
   else
      $y_{\mathcal{C}} = \mathrm{ML}(\tilde{Q}_{\mathcal{C}}, x_{\mathcal{C}}, \mathcal{C}, \gamma)$;
      $y_{\mathcal{C}} = \mathrm{ML}(\tilde{Q}_{\mathcal{C}}, y_{\mathcal{C}}, \mathcal{C}, \gamma)$;
   $y_{\mathcal{D}} = y_{\mathcal{C}} P_{x'_{\mathcal{D}}}$;
   $y'_{\mathcal{D}} = S(\tilde{Q}_{\mathcal{D}}, y_{\mathcal{D}}, w, \nu_2)$;
   if ($C == F$) then                                                   (step 4)
      $\gamma = 2$;
   $x_{\mathcal{D}} = y'_{\mathcal{D}}$; $it = it + 1$;                                       (step 5)
   $x_{\mathcal{D}} = x_{\mathcal{D}}/(x_{\mathcal{D}} e)$; $r = -x_{\mathcal{D}} \tilde{Q}_{\mathcal{D}}$;                                   (step 6)
   if ($it \geq MAX\_IT$ or $time \geq MAX\_TIME$ or $\|r\| \leq STOP\_TOL$) then   (step 7)
      $stop = TRUE$;
   else if ($O == DYNAMIC$) then                                        (step 8)
      sort LLM indices $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$ into increasing order of $\|r_k\|$,
      where $r_k$ is the residual associated with LLM $k$ and is computed from $r$;
   else if ($O == CIRCULAR$) then
      $\mathcal{D}_k = \mathcal{D}_{(k \bmod K)+1}$ for $k = 1, 2, \ldots, K$;
until($stop$);
take $x_{\mathcal{D}}$ as the steady state vector $\pi$ of the HMM;
```

Algorithm 2. [illegible] $\mathcal{D}$.

```
function ML($\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}}, \mathcal{D}, \gamma$)
if ($|\mathcal{D}| == 1$) then
   $y'_{\mathcal{D}}$ = solve($\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}}$) subject to $y'_{\mathcal{D}} e = 1$;                     (step 1)
```

```
    if ($C == F$) then                                          (step 2)
        $\gamma = 1$;
else
    $x'_{\mathcal{D}} = S(\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}}, w, \nu_1)$;          (step 3)
    remove $\mathcal{D}_1$ from $\mathcal{D}$ by aggregation to give $\mathcal{C}$;        (step 4)
    $\tilde{Q}_{\mathcal{C}} = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} R_{\mathcal{D}}$; $x_{\mathcal{C}} = x'_{\mathcal{D}} R_{\mathcal{D}}$;        (step 5)
    if ($\gamma == 1$) then                                     (step 6)
        $y_{\mathcal{C}} = \mathrm{ML}(\tilde{Q}_{\mathcal{C}}, x_{\mathcal{C}}, \mathcal{C}, \gamma)$;
    else
        $y_{\mathcal{C}} = \mathrm{ML}(\tilde{Q}_{\mathcal{C}}, x_{\mathcal{C}}, \mathcal{C}, \gamma)$;
        $y_{\mathcal{C}} = \mathrm{ML}(\tilde{Q}_{\mathcal{C}}, y_{\mathcal{C}}, \mathcal{C}, \gamma)$;
    $y_{\mathcal{D}} = y_{\mathcal{C}} P_{x'_{\mathcal{D}}}$;                                   (step 7)
    $y'_{\mathcal{D}} = S(\tilde{Q}_{\mathcal{D}}, y_{\mathcal{D}}, w, \nu_2)$;          (step 8)
return($y'_{\mathcal{D}}$);
```

---

The variable $\gamma$ in the two algorithms determines the number of recursive calls to the ML function. It is initialized to 2 for a W- or an F-cycle and to 1 for a V-cycle before ML starts executing for the first time. After this point, there are two places where the value of $\gamma$ changes, and these happen only for an F-cycle. Hence, for a V-cycle $\gamma$ remains 1, and for a W-cycle it remains 2, meaning for V- and W-cycles 1 and 2 recursive calls, respectively, are made to the ML function on the next coarser grid. On the other hand, for an F-cycle $\gamma$ is set to 1 at the boundary case of the recursion (see step 2 in Algorithm 2). Hence, an F-cycle can be seen as a recursive call to a W-cycle followed by a recursive call to a V-cycle. After the F-cycle is over, $\gamma$ is reset to 2 in step 4 of Algorithm 1 so as to be ready for a new ML iteration [33, pp. 174–175].

Each ML iteration starts and ends with some number of iterations using the smoother $S$. See respectively the two statements after step 3 and before step 4 in Algorithm 1. The same is true for each execution of the recursive ML function at intermediate grids, as can be seen in steps 3 and 8 of Algorithm 2. The first two arguments of the call to $S$ in both algorithms represent the grid to be used in the smoothing process and the vector to be smoothed. The parameter $\omega$ in the call to $S$ is the relaxation parameter for JOR and SOR. Although the user can be given the flexibility to change the numbers of pre- and postsmoothings in the two algorithms, depending on the residual norms (see Algorithms 1 and 2 in [13]), we consider $\nu_1$ pre- and $\nu_2$ postsmoothings at each level in order to simplify the description of the algorithms in this presentation.

The order of aggregating LLMs in each ML iteration is determined by the list $\mathcal{D}$ defined in Algorithm 1. The elements of $\mathcal{D}$ from its head to its tail are denoted respectively by $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{K+1}$. The subscripts of these elements indicate their positions in $\mathcal{D}$. In each ML iteration, the HLM is always the last model to be handled due to its special position in the hierarchy. Hence, $\mathcal{D}_{K+1}$ is given the value $(K+1)$ and is associated with the HLM; the tail of $\mathcal{D}$ always has this value and does not change. Initially, LLM $k$ is associated with element $\mathcal{D}_k$, which has the value $k$ for $k = 1, 2, \ldots, K$ (see step 1 of Algorithm 1). In each ML iteration, LLMs are aggregated according to these values starting from the element at the head of the list (see the second statement in the repeat-until loop of Algorithm 1). Hence, LLM $\mathcal{D}_1$ is the first LLM to be aggregated.

In the *FIXED* order of aggregating LLMs, the initial assignment of values to the elements of $\mathcal{D}$ does not change after the ML method starts executing; this is the default order. In the *CIRCULAR* order, at the end of each ML iteration a circular shift of elements $\mathcal{D}_1$ through $\mathcal{D}_K$ in the list is performed; this ensures some kind of fairness in aggregating LLMs in the next ML iteration. On the other hand, the *DYNAMIC* order sorts the elements $\mathcal{D}_1$ through $\mathcal{D}_K$ according to the residual norms mapped (or restricted) to the corresponding LLM at the end of the ML iteration, and aggregates the LLMs in this sorted order in the next ML iteration (see step 8 of Algorithm 1). This ensures that LLMs which have smaller residual norms are aggregated earlier at finer grids. We expect small residual norms to be indicative of good approximations in those LLMs. Note that at each intermediate grid the recursive ML function is invoked for the next coarser grid with the list of LLMs in $\mathcal{C}$, which is formed by removing the LLM at the head of the incoming list $\mathcal{D}$ (i.e., $\mathcal{D}_1$) by aggregation (see step 4 in Algorithm 2). Once the list of LLMs is exhausted, that is $(K+1)$ is the only value remaining in list $\mathcal{D}$, backtracking from recursion starts by solving a linear system as large as the HLM matrix (see step 1 in Algorithm 2). This is indicated by the call to the function [illegible], which takes the coarsest grid $\tilde{Q}_{\mathcal{D}}$ as input and produces the solution $y'_{\mathcal{D}}$ up to machine precision directly (i.e., by Gaussian elimination) if $|\mathcal{S}^{(K+1)}|$ is relatively small, else iteratively using the smoother $S$ and the current approximation $x_{\mathcal{D}}$ as the starting vector.

The ML solver starts with $x_{\mathcal{D}}$, which is usually set to the uniform distribution, and $r$ as the corresponding residual vector. The repeat-until loop increments the number of ML iterations denoted by *it* and continues until *it* reaches the maximum number of iterations in *MAX_IT*, solution time reaches *MAX_TIME*, or the residual $r$ reaches the user-defined *STOP_TOL*. We remark that the smoothers of choice require two vectors of length $n$ and two vectors (three in SOR) as long as the maximum number of microstates per macrostate in the HMM. One of the vectors of length $n$ in SOR is required for the computation of residuals in the implementation of *DYNAMIC* ordering of LLMs for aggregation. Furthermore, if one turns off the call(s) in Algorithm 1 to Algorithm 2, Algorithm 1 reduces to an iterative solver in which $(\nu_1 + \nu_2)$ iterations are performed on $Q$ with the iterative method $S$ at each ML cycle. This is a useful feature for debugging.

**3.2. Operators and implementation.** Before we discuss the operation that computes the next coarser grid $\tilde{Q}_{\mathcal{C}}$ from the grid $\tilde{Q}_{\mathcal{D}}$ using the smoothed vector $x'_{\mathcal{D}}$ (see step 5 in Algorithm 2), let us define the state spaces of the grids used in the ML method for large sparse MCs in terms of a mapping [30, pp. 192–197].

DEFINITION 3.1. [illegible] $\mathcal{S}_{\mathcal{D}}$ [illegible] $\mathcal{S}_{\mathcal{C}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{C}}$ [illegible] $f_{\mathcal{D}} : \mathcal{S}_{\mathcal{D}} \longrightarrow \mathcal{S}_{\mathcal{C}}$ [illegible] $\mathcal{S}_{\mathcal{D}}$ [illegible] $\mathcal{S}_{\mathcal{C}}$

The mapping $f_{\mathcal{D}}$ is surjective (i.e., onto); it satisfies

$$\exists s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}},\ f_{\mathcal{D}}(s_{\mathcal{D}}) = s_{\mathcal{C}} \quad \text{for each} \quad s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$$

and $|\mathcal{S}_{\mathcal{C}}| \leq |\mathcal{S}_{\mathcal{D}}|$. When $|\mathcal{S}_{\mathcal{C}}| = |\mathcal{S}_{\mathcal{D}}|$, the mapping becomes bijective (i.e., one-to-one onto). From Definition 3.1 and [30, p. 179], we have the next proposition.

PROPOSITION 3.2. [illegible] $\tilde{f}_{\mathcal{D}}$ [illegible] $f_{\mathcal{D}}$ [illegible] $\tilde{f}_{\mathcal{D}}$ [illegible] $\mathcal{S}_{\mathcal{C}}$ [illegible] $\mathcal{S}_{\mathcal{D}}$ [illegible] $|\mathcal{S}_{\mathcal{C}}| = |\mathcal{S}_{\mathcal{D}}|$ [illegible] $f_{\mathcal{D}}$ [illegible]

Proposition 3.2 says that, if there is at least one state in $\mathcal{S}_{\mathcal{C}}$ to which multiple states from $\mathcal{S}_{\mathcal{D}}$ are mapped under $f_{\mathcal{D}}$ (i.e., $|\mathcal{S}_{\mathcal{C}}| < |\mathcal{S}_{\mathcal{D}}|$), then the converse of $f_{\mathcal{D}}$ cannot be a function; it is just a relation.

For HMMs, the Kronecker structure (see Definition 2.2 and Proposition 2.4) and the order of component aggregation determine $\mathcal{S}_{\mathcal{D}}$ and $\mathcal{S}_{\mathcal{C}}$ as in the next proposition.

PROPOSITION 3.3. [illegible] 1 [illegible] 2 [illegible] $\mathcal{D}$ [illegible] $\mathcal{C}$ [illegible] $\mathcal{S}_\mathcal{D}$ [illegible] $\mathcal{S}_\mathcal{C}$ [illegible]

$$\mathcal{S}_\mathcal{D} = \bigcup_{j \in \mathcal{S}^{(K+1)}} \times_{k=1}^{|\mathcal{D}|} \mathcal{S}_j^{(\mathcal{D}_k)} \quad [illegible] \quad \mathcal{S}_\mathcal{C} = \bigcup_{j \in \mathcal{S}^{(K+1)}} \times_{k=1}^{|\mathcal{C}|} \mathcal{S}_j^{(\mathcal{C}_k)},$$

[illegible] $\times$ [illegible]

$$|\mathcal{S}_\mathcal{D}| = \sum_{j=0}^{|\mathcal{S}^{(K+1)}|-1} \prod_{k=1}^{|\mathcal{D}|} |\mathcal{S}_j^{(\mathcal{D}_k)}| \quad [illegible] \quad |\mathcal{S}_\mathcal{C}| = \sum_{j=0}^{|\mathcal{S}^{(K+1)}|-1} \prod_{k=1}^{|\mathcal{C}|} |\mathcal{S}_j^{(\mathcal{C}_k)}|.$$

[illegible] 1 $|\mathcal{S}_\mathcal{D}| = n$

Observe from Definition 2.2 that $\mathcal{S}_\mathcal{D}$ and $\mathcal{S}_\mathcal{C}$ for HMMs given in Proposition 3.3 satisfy the mapping $f_\mathcal{D} : \mathcal{S}_\mathcal{D} \longrightarrow \mathcal{S}_\mathcal{C}$ in Definition 3.1.

Now we return to the computation of the coarser grid and the coarser approximation. For each state $s_\mathcal{C} \in \mathcal{S}_\mathcal{C}$, the columns of the grid $\tilde{Q}_\mathcal{D}$ corresponding to the states in $\mathcal{S}_\mathcal{D}$ that get mapped to the same state $s_\mathcal{C}$ are summed. The aggregation on the columns of $\tilde{Q}_\mathcal{D}$ is also performed on the columns of the smoothed row vector $x_\mathcal{D}^{'}$ yielding the vector $x_\mathcal{C}$ in step 5 of Algorithm 2. These are achieved by using the [illegible] [25] (or aggregation) operator defined next.

DEFINITION 3.4. [illegible] $(|\mathcal{S}_\mathcal{D}| \times |\mathcal{S}_\mathcal{C}|)$ [illegible] $R_\mathcal{D}$ [illegible] $f_\mathcal{D} : \mathcal{S}_\mathcal{D} \longrightarrow \mathcal{S}_\mathcal{C}$ [illegible] $(s_\mathcal{D}, s_\mathcal{C})$ [illegible]

$$r_\mathcal{D}(s_\mathcal{D}, s_\mathcal{C}) = \begin{cases} 1 & [illegible]\ f_\mathcal{D}(s_\mathcal{D}) = s_\mathcal{C}, \\ 0 & [illegible], \end{cases} \qquad [illegible]\ s_\mathcal{D} \in \mathcal{S}_\mathcal{D}\ [illegible]\ s_\mathcal{C} \in \mathcal{S}_\mathcal{C}.$$

PROPOSITION 3.5. [illegible] $R_\mathcal{D}$ [illegible] 1 [illegible] 1 [illegible] $R_\mathcal{D}$ [illegible] $\mathrm{rank}(R_\mathcal{D}) = |\mathcal{S}_\mathcal{C}|$ [illegible] $\tilde{Q}_\mathcal{D} R_\mathcal{D}$ [illegible] $\tilde{Q}_\mathcal{D}$ [illegible]

For each state $s_\mathcal{C} \in \mathcal{S}_\mathcal{C}$, the rows of $\tilde{Q}_\mathcal{D} R_\mathcal{D}$ corresponding to the states in $\mathcal{S}_\mathcal{D}$ that are mapped to the same state $s_\mathcal{C}$ are multiplied with the corresponding normalized elements of the smoothed row vector $x_\mathcal{D}^{'}$ and summed. This is achieved by using the [illegible] [25] (or disaggregation) operator defined next.

DEFINITION 3.6. [illegible] $(|\mathcal{S}_\mathcal{C}| \times |\mathcal{S}_\mathcal{D}|)$ [illegible] $P_{x_\mathcal{D}^{'}}$ [illegible] $f_\mathcal{D} : \mathcal{S}_\mathcal{D} \longrightarrow \mathcal{S}_\mathcal{C}$ [illegible] $(s_\mathcal{C}, s_\mathcal{D})$ [illegible]

$$p_{x_\mathcal{D}^{'}}(s_\mathcal{C}, s_\mathcal{D}) = \begin{cases} x_\mathcal{D}^{'}(s_\mathcal{D}) / \sum_{s_\mathcal{D} \in \mathcal{S}_\mathcal{D}, f_\mathcal{D}(s_\mathcal{D}) = s_\mathcal{C}} x_\mathcal{D}^{'}(s_\mathcal{D}) & [illegible]\ f_\mathcal{D}(s_\mathcal{D}) = s_\mathcal{C}, \\ 0 & [illegible], \end{cases}$$

$$[illegible]\ s_\mathcal{D} \in \mathcal{S}_\mathcal{D}\ [illegible]\ s_\mathcal{C} \in \mathcal{S}_\mathcal{C}.$$

PROPOSITION 3.7. [illegible] $x_\mathcal{D}^{'} > 0$ [illegible] $P_{x_\mathcal{D}^{'}}$ [illegible] $R_\mathcal{D}$ [illegible] $\mathrm{rank}(P_{x_\mathcal{D}^{'}}) = |\mathcal{S}_\mathcal{C}|$ [illegible] $x_\mathcal{D}^{'} > 0$ [illegible] $P_{x_\mathcal{D}^{'}}$ [illegible] $P_{x_\mathcal{D}^{'}}$ [illegible] 1 [illegible] $R_\mathcal{D}$ [illegible] $\tilde{Q}_\mathcal{D} R_\mathcal{D}$ [illegible] $P_{x_\mathcal{D}^{'}}$ [illegible] $(|\mathcal{S}_\mathcal{C}| \times |\mathcal{S}_\mathcal{C}|)$ [illegible] $\tilde{Q}_\mathcal{C}$ [illegible] $x_\mathcal{D}^{'}$

The prolongation operator depends not only on $\mathcal{S}_{\mathcal{D}}$ and $\mathcal{S}_{\mathcal{C}}$, but also on the smoothed vector $x'_{\mathcal{D}}$, which is indicated by using the subscript $x'_{\mathcal{D}}$ rather than $\mathcal{D}$. This implies that the elements of $\tilde{Q}_{\mathcal{C}}$ depend on $x'_{\mathcal{D}}$ and will be different in each iteration of the ML solver.

LEMMA 3.8. [illegible] $x'_{\mathcal{D}} > 0$ [illegible] $P_{x'_{\mathcal{D}}} R_{\mathcal{D}} = I_{\mathcal{C}}$ [illegible] $I_{\mathcal{C}}$ [illegible] $|\mathcal{S}_{\mathcal{C}}|$

[illegible]. The identity follows from Propositions 3.5 and 3.7 by the facts that $P_{x'_{\mathcal{D}}} \geq 0$, $R_{\mathcal{D}} \geq 0$, $P_{x'_{\mathcal{D}}}$ has the same nonzero structure as $R^T_{\mathcal{D}}$, $P_{x'_{\mathcal{D}}} e = e$, and $e^T R^T_{\mathcal{D}} = e^T$. ☐

When $x'_{\mathcal{D}} > 0$, we can state the next corollary [23, p. 387] using $R_{\mathcal{D}}(P_{x'_{\mathcal{D}}} R_{\mathcal{D}})P_{x'_{\mathcal{D}}} = R_{\mathcal{D}}(I_{\mathcal{C}})P_{x'_{\mathcal{D}}} = R_{\mathcal{D}}P_{x'_{\mathcal{D}}}$ from Lemma 3.8, $R_{\mathcal{D}} \geq 0$, $R_{\mathcal{D}}e = e$ and $P_{x'_{\mathcal{D}}} \geq 0$, $P_{x'_{\mathcal{D}}} e = e$ from Propositions 3.5 and 3.7, respectively.

COROLLARY 3.9. [illegible] $x'_{\mathcal{D}} > 0$ [illegible] $(|\mathcal{S}_{\mathcal{D}}| \times |\mathcal{S}_{\mathcal{D}}|)$ [illegible]

$$H_{x'_{\mathcal{D}}} = R_{\mathcal{D}} P_{x'_{\mathcal{D}}}$$

[illegible] $H_{x'_{\mathcal{D}}} \geq 0$ [illegible] $H^2_{x'_{\mathcal{D}}} = H_{x'_{\mathcal{D}}}$ [illegible] $H_{x'_{\mathcal{D}}} e = e$

LEMMA 3.10. [illegible] $x'_{\mathcal{D}} > 0$ [illegible] $x'_{\mathcal{D}} H_{x'_{\mathcal{D}}} = x'_{\mathcal{D}}$

[illegible]. The identity follows from the definitions of restriction and prolongation operations (see Definitions 3.4 and 3.6) and the fact that the restricted and then prolonged row vector is $x'_{\mathcal{D}}$. ☐

The analysis in section 4 is based on showing that the coarser grid $\tilde{Q}_{\mathcal{C}}$ is an irreducible CTMC and $x_{\mathcal{C}} > 0$ if the finer grid $\tilde{Q}_{\mathcal{D}}$ is an irreducible CTMC and $x'_{\mathcal{D}} > 0$. This has been done for HMMs with one macrostate in [9, p. 348]. In section 4, we show the results for the mapping $f : \mathcal{S}_{\mathcal{D}} \longrightarrow \mathcal{S}_{\mathcal{C}}$ in Definition 3.1.

Step 7 in Algorithm 2 corresponds to the opposite of what is done on $x'_{\mathcal{D}}$ in step 5; that is, it performs disaggregation using the newly computed vector $y_{\mathcal{C}}$ and the prolongation operator $P_{x'_{\mathcal{D}}}$ (which is based on the smoothed vector $x'_{\mathcal{D}}$) to obtain the vector $y_{\mathcal{D}}$. The next result follows from Proposition 3.7

PROPOSITION 3.11. [illegible] $y_{\mathcal{C}} > 0$ [illegible] $x'_{\mathcal{D}} > 0$ [illegible] $y_{\mathcal{D}} = y_{\mathcal{C}} P_{x'_{\mathcal{D}}} > 0$ [illegible] $e^T P_{x'_{\mathcal{D}}} > 0$

Similar aggregation and disaggregation operations are performed in Algorithm 1 at the finest grid $Q$.

The Kronecker representation of $\tilde{Q}_{\mathcal{C}}$ for an HMM with one macrostate is given in [9, p. 347]. Here we extend it to multiple macrostates and show that $\tilde{Q}_{\mathcal{C}}$ can be expressed as a sum of Kronecker products as in Definition 2.2 using $\sum_{i,j \in \mathcal{S}^{(K+1)}} |\mathcal{T}_{i,j}|$ vectors each of length at most $\max_{j \in \mathcal{S}^{(K+1)}} (\prod_{k=2}^{|\mathcal{C}|} |\mathcal{S}_j^{(\mathcal{C}_k)}|)$ and the matrices corresponding to the components in $\mathcal{C}$ excluding $(K+1)$, which denotes the HLM (see Proposition 3.3). More specifically, we have the next definition.

DEFINITION 3.12. [illegible] $h = \mathcal{D}_1$ [illegible] $s_{\mathcal{C}}$ [illegible] $t_e$ [illegible] $(i,j)$ [illegible] $\tilde{Q}_{\mathcal{C}}$ [illegible]

$$a_{(\mathcal{C},t_e),(i,j)}(s_{\mathcal{C}}) = \frac{\left(\sum_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}, f_{\mathcal{D}}(s_{\mathcal{D}}) = s_{\mathcal{C}}} x'_{\mathcal{D}}(s_{\mathcal{D}})\ a_{(\mathcal{D},t_e),(i,j)}(s_{\mathcal{D}})\ (e^T_{s_{\mathcal{D}}(h)} Q^{(h)}_{t_e}(\mathcal{S}^{(h)}_i, \mathcal{S}^{(h)}_j)e)\right)}{x_{\mathcal{C}}(s_{\mathcal{C}})}$$

$$[illegible]\ s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}},\ t_e \in \mathcal{T}_{i,j},\ [illegible]\ i,j \in \mathcal{S}^{(K+1)},$$

[illegible] $a_{(\mathcal{D},t_e),(i,j)} = e$ [illegible] $\mathcal{D}$ [illegible] $s_{\mathcal{D}}(h) \in \mathcal{S}^{(h)}$ [illegible] $e_{s_{\mathcal{D}}(h)}$ [illegible] $s_{\mathcal{D}}(h)$ [illegible] $|\mathcal{S}_i^{(h)}|$

With this definition, blocks of the matrix $\tilde{Q}_{\mathcal{C}}$ become

$$\begin{aligned}
\tilde{Q}_{\mathcal{C}}(j,j) = {} & \bigoplus_{k=1}^{|\mathcal{C}|-1} Q_{t_0}^{(\mathcal{C}_k)}(\mathcal{S}_j^{(\mathcal{C}_k)}, \mathcal{S}_j^{(\mathcal{C}_k)}) + \sum_{t_e \in \mathcal{T}_{j,j}} \bigotimes_{k=1}^{|\mathcal{C}|-1} \operatorname{diag}(a_{(\mathcal{C},t_e),(j,j)}) Q_{t_e}^{(\mathcal{C}_k)}(\mathcal{S}_j^{(\mathcal{C}_k)}, \mathcal{S}_j^{(\mathcal{C}_k)}) \\
& - \bigoplus_{k=1}^{|\mathcal{C}|-1} \operatorname{diag}(Q_{t_0}^{(\mathcal{C}_k)}(\mathcal{S}_j^{(\mathcal{C}_k)}, \mathcal{S}_j^{(\mathcal{C}_k)}) e) \\
& - \sum_{i \in \mathcal{S}^{(K+1)}} \sum_{t_e \in \mathcal{T}_{j,i}} \bigotimes_{k=1}^{|\mathcal{C}|-1} \operatorname{diag}(a_{(\mathcal{C},t_e),(j,i)}) \operatorname{diag}(Q_{t_e}^{(\mathcal{C}_k)}(\mathcal{S}_j^{(\mathcal{C}_k)}, \mathcal{S}_i^{(\mathcal{C}_k)}) e) \\
& \qquad \text{for } j \in \mathcal{S}^{(K+1)}, \\
\tilde{Q}_{\mathcal{C}}(i,j) = {} & \sum_{t_e \in \mathcal{T}_{i,j}} \bigotimes_{k=1}^{|\mathcal{C}|-1} \operatorname{diag}(a_{(\mathcal{C},t_e),(i,j)}) Q_{t_e}^{(\mathcal{C}_k)}(\mathcal{S}_i^{(\mathcal{C}_k)}, \mathcal{S}_j^{(\mathcal{C}_k)}) \quad \text{for } i,j \in \mathcal{S}^{(K+1)}, i \neq j.
\end{aligned}$$

Observe from Proposition 2.3 that the last two terms of $\tilde{Q}_{\mathcal{C}}(j,j)$ return a diagonal matrix which sums the rows of $\tilde{Q}_{\mathcal{C}}(j,j)$ to zero. Furthermore, the vectors $a_{(\mathcal{D},t_e),(i,j)}$ for $t_e \in \mathcal{T}_{i,j}$ and $i,j \in \mathcal{S}^{(K+1)}$ at the finest level consist of all 1's, and therefore need not be stored. When the recursion ends at the HLM, $\tilde{Q}_{\mathcal{C}}$ is a $(|\mathcal{S}^{(K+1)}| \times |\mathcal{S}^{(K+1)}|)$ CTMC, and therefore is generated and stored explicitly in sparse format so that it can be solved either directly or iteratively, as we discussed. We remark that $a_{(\mathcal{C},t_e),(i,j)} = e$ for those $t_e$ which have all $Q_{t_e}^{(\mathcal{C}_k)}(\mathcal{S}_i^{(\mathcal{C}_k)}, \mathcal{S}_j^{(\mathcal{C}_k)})$ as diagonal matrices of size $(|\mathcal{S}_i^{(\mathcal{C}_k)}| \times |\mathcal{S}_j^{(\mathcal{C}_k)}|)$ with 1's along their diagonal for $k = 1, 2, \ldots, |\mathcal{C}| - 1$ and $i,j \in \mathcal{S}^{(K+1)}$. Since component matrices forming $\tilde{Q}_{\mathcal{C}}(i,j)$ for $i,j \in \mathcal{S}^{(K+1)}, i \neq j$, can very well be rectangular, we refrain from using $I$, and remark that such vectors need not be stored either.

The next section presents results on the convergence of the proposed class of ML methods for large sparse MCs.

## 4. Convergence of ML methods.

Convergence analysis of AMG with a post-smoother of the Richardson relaxation type (see [26, p. 412]) and a two-level grid for symmetric positive definite linear systems arising from finite element approximations to a particular differential operator appears in [18]. Therein, it is shown that the convergence rate of the method is independent of the problem size when the relaxation parameter of the smoother is chosen appropriately [18, p. 480]. On the other hand, [27] casts AMG as a special case of multi-iterative methods for positive definite linear systems in which two or more iterative techniques are successively used in each iteration to improve the error in different subspaces. When the method is AMG, one of these multi-iterative methods has an iteration matrix associated with the coarse grid correction. A convergence analysis for a two-level grid with a Richardson iteration as the presmoother and a prolongation operator with (block) antidiagonal structure is provided. Using information about the eigenvalues of the coefficient matrix together with the particular smoother, it is shown that the AMG method possesses a convergence rate independent of the problem size for banded (block) Toeplitz matrices. Although the *POWER* smoother used by the proposed class of ML methods is also a Richardson relaxation, as will be shown in this section, the methods are geared towards CTMCs, which have different characteristics. Recently, in [22] the results in

[21] are improved, and an asymptotic convergence result is provided for a two-level IAD method which uses postsmoothings of the POWER type. However, fast convergence cannot be guaranteed in a general setting even when there are only two levels [22, p. 340]. Hence, the results in the next subsections should be received as a step towards improving the formulation and understanding the convergence behavior of the proposed class of ML methods.

Let $\mathcal{D}$ represent the current level and $\mathcal{C}$ represent the next coarser level in the ML iteration, as in Algorithms 1 and 2. Let $\mathcal{S}_\mathcal{D}$ and $\mathcal{S}_\mathcal{C}$ denote respectively the state spaces of $\tilde{Q}_\mathcal{D}$ and $\tilde{Q}_\mathcal{C}$, and assume that the mapping of states from $\mathcal{S}_\mathcal{D}$ to the states in $\mathcal{S}_\mathcal{C}$ is onto and satisfies $|\mathcal{S}_\mathcal{C}| \leq |\mathcal{S}_\mathcal{D}|$ as in Definition 3.1. The results that are presented in this section for Algorithms 1 and 2 are general in that the Kronecker representation of the grids particular to HMMs is not utilized.

**4.1. Irreducibility of the coarser grids.** Recall that $R_\mathcal{D} \geq 0$, $R_\mathcal{D} e = e$, $e^T R_\mathcal{D} > 0$ from Proposition 3.5, and if $x^{'}_\mathcal{D} > 0$, then $P_{x^{'}_\mathcal{D}} \geq 0$, $P_{x^{'}_\mathcal{D}} e = e$, $e^T P_{x^{'}_\mathcal{D}} > 0$ from Proposition 3.7. Now, consider the definition of irreducibility given in [23, p. 209] and [29, p. 13]. Then the following lemma, which will be used to discuss the convergence of the ML method, can be proved.

Lemma 4.1. [illegible] $\tilde{Q}_\mathcal{C} = P_{x^{'}_\mathcal{D}} \tilde{Q}_\mathcal{D} R_\mathcal{D}$ [illegible] $x_\mathcal{C} = x^{'}_\mathcal{D} R_\mathcal{D} > 0$ [illegible] $\tilde{Q}_\mathcal{D}$ [illegible] $x^{'}_\mathcal{D} > 0$

[illegible] First, we show that $\tilde{Q}_\mathcal{C} = P_{x^{'}_\mathcal{D}} \tilde{Q}_\mathcal{D} R_\mathcal{D}$ is an irreducible CTMC. Without losing generality, consider the pair of different states $s_\mathcal{D}, s'_\mathcal{D} \in \mathcal{S}_\mathcal{D}$. Through $f : \mathcal{S}_\mathcal{D} \longrightarrow \mathcal{S}_\mathcal{C}$ in Definition 3.1, this pair of states are mapped respectively to the states $s_\mathcal{C}, s'_\mathcal{C} \in \mathcal{S}_\mathcal{C}$ (i.e., $f(s_\mathcal{D}) = s_\mathcal{C}$ and $f(s'_\mathcal{D}) = s'_\mathcal{C}$). Since $\tilde{Q}_\mathcal{D}$ is irreducible, there exists a path of transitions from $s_\mathcal{D}$ to $s'_\mathcal{D}$ in $\mathcal{S}_\mathcal{D}$ in the form $s_\mathcal{D} = s_1, s_2, \ldots, s_m = s'_\mathcal{D}$, where $m \leq |\mathcal{S}_\mathcal{D}|$, $s_k \in \mathcal{S}_\mathcal{D}$, and $\tilde{q}_D(s_k, s_{k+1}) > 0$ for $k \in \{1, 2, \ldots, m-1\}$. Mapping this path onto $\mathcal{S}_\mathcal{C}$ yields the path $s_\mathcal{C} = t_1, t_2, \ldots, t_m = s'_\mathcal{C}$, where $f(s_k) = t_k \in \mathcal{S}_\mathcal{C}$. Now, let $e_{t_k}$ denote the $t_k$th column of $I_\mathcal{C}$. Then, in the mapped path, we either have $t_k = t_{k+1}$ or $\tilde{q}_\mathcal{C}(t_k, t_{k+1}) > 0$, where the latter follows from

$$\begin{aligned}\tilde{q}_\mathcal{C}(t_k, t_{k+1}) &= e^T_{t_k} \tilde{Q}_\mathcal{C} e_{t_{k+1}} \\ &= (e^T_{t_k} P_{x^{'}_\mathcal{D}}) \tilde{Q}_\mathcal{D} (R_\mathcal{D} e_{t_{k+1}}) \geq p_{x^{'}_\mathcal{D}}(t_k, s_k) \tilde{q}_\mathcal{D}(s_k, s_{k+1}) r_\mathcal{D}(s_{k+1}, t_{k+1}),\end{aligned}$$

since $x_D(s_k) > 0$ (implying $p_{x^{'}_\mathcal{D}}(t_k, s_k) > 0$ from Definition 3.6), $\tilde{q}_\mathcal{D}(s_k, s_{k+1}) > 0$, and $f(s_{k+1}) = t_{k+1}$ (implying $r_\mathcal{D}(s_{k+1}, t_{k+1}) = 1$ from Definition 3.4). Thus we conclude that $s'_\mathcal{C}$ is reachable from $s_\mathcal{C}$.

We have effectively shown that each state in $\tilde{Q}_\mathcal{C}$ is reachable from every other state. The question that arises at this point is whether a row of $\tilde{Q}_\mathcal{C}$ can become zero after the restriction. The answer is no, as long as $\mathcal{S}_\mathcal{C}$ has multiple states (i.e., $|\mathcal{S}_\mathcal{C}| > 1$), since all states in $\mathcal{S}_\mathcal{D}$ that are mapped to a particular state in $\mathcal{S}_\mathcal{C}$ cannot have all their transitions among themselves. This would imply that $\tilde{Q}_\mathcal{D}$ is reducible, which is a contradiction. Furthermore, since the row sums of $\tilde{Q}_\mathcal{C}$ are zero (i.e., $\tilde{Q}_\mathcal{C} e = (P_{x^{'}_\mathcal{D}} \tilde{Q}_\mathcal{D} R_\mathcal{D}) e = P_{x^{'}_\mathcal{D}} \tilde{Q}_\mathcal{D} (R_\mathcal{D} e) = P_{x^{'}_\mathcal{D}} \tilde{Q}_\mathcal{D} e = 0$ because $\tilde{Q}_\mathcal{D}$ is a CTMC and $\tilde{Q}_\mathcal{D} e = 0$), its diagonal must be equal to its negated off-diagonal row sums. Hence, $\tilde{Q}_\mathcal{C}$ is an irreducible CTMC.

Now we show that $x_\mathcal{C} > 0$. Since $x_\mathcal{C} = x^{'}_\mathcal{D} R_\mathcal{D}$, $x^{'}_\mathcal{D} = e^T \mathrm{diag}(x^{'}_\mathcal{D})$, where $\mathrm{diag}(x^{'}_\mathcal{D})$ is the diagonal matrix with $x^{'}_\mathcal{D}$ along its diagonal, $\mathrm{diag}(x^{'}_\mathcal{D}) R_\mathcal{D}$ has the same nonzero structure as $R_\mathcal{D}$, and $e^T R_\mathcal{D} > 0$, we have $x_\mathcal{C} = x^{'}_\mathcal{D} R_\mathcal{D} = (e^T \mathrm{diag}(x^{'}_\mathcal{D})) R_\mathcal{D} = e^T (\mathrm{diag}(x^{'}_\mathcal{D}) R_\mathcal{D}) > 0$ when $x^{'}_\mathcal{D} > 0$. $\square$

Corollary 4.2. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $x'_{\mathcal{D}} > 0$ [illegible] $x'_{\mathcal{D}}\tilde{Q}_{\mathcal{D}} = 0$ [illegible] $x_{\mathcal{C}}\tilde{Q}_{\mathcal{C}} = 0$ [illegible] $\tilde{Q}_{\mathcal{C}} = P_{x'_{\mathcal{D}}}\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}}$ [illegible] $x_{\mathcal{C}} = x'_{\mathcal{D}}R_{\mathcal{D}}$

[illegible] We have $x_{\mathcal{C}}\tilde{Q}_{\mathcal{C}} = (x'_{\mathcal{D}}R_{\mathcal{D}})(P_{x'_{\mathcal{D}}}\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}}) = (x'_{\mathcal{D}}R_{\mathcal{D}}P_{x'_{\mathcal{D}}})\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}} = (x'_{\mathcal{D}}H_{x'_{\mathcal{D}}})$ $\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}} = (x'_{\mathcal{D}})\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}} = (x'_{\mathcal{D}}\tilde{Q}_{\mathcal{D}})R_{\mathcal{D}} = 0$, since $x'_{\mathcal{D}}H_{x'_{\mathcal{D}}} = x'_{\mathcal{D}}$ from Lemma 3.10 and $x'_{\mathcal{D}}\tilde{Q}_{\mathcal{D}} = 0$ by assumption. □

Proposition 4.3. [illegible] $\pi_{\mathcal{D}} = \pi > 0$ [illegible] $Q_{\mathcal{D}} = Q$ [illegible] $\mathcal{D}$ [illegible] $\mathcal{C}$ [illegible] $Q_{\mathcal{C}} = P_{\pi_{\mathcal{D}}}Q_{\mathcal{D}}R_{\mathcal{D}}$ [illegible] $\pi_{\mathcal{C}} = \pi_{\mathcal{D}}R_{\mathcal{D}} > 0$ [illegible] $\mathcal{D}$ [illegible] $\mathcal{C}$ [illegible] $\mathcal{D}$ [illegible] $Q_{\mathcal{D}}$ [illegible] $\pi_{\mathcal{D}}$ [illegible] $Q_{\mathcal{C}}$ [illegible] $\mathcal{C}$

The proposition follows from $\pi_{\mathcal{C}}Q_{\mathcal{C}} = (\pi_{\mathcal{D}}R_{\mathcal{D}})(P_{\pi_{\mathcal{D}}}Q_{\mathcal{D}}R_{\mathcal{D}}) = (\pi_{\mathcal{D}}R_{\mathcal{D}}P_{\pi_{\mathcal{D}}})$ $Q_{\mathcal{D}}R_{\mathcal{D}} = (\pi_{\mathcal{D}}H_{\pi_{\mathcal{D}}})Q_{\mathcal{D}}R_{\mathcal{D}} = (\pi_{\mathcal{D}})Q_{\mathcal{D}}R_{\mathcal{D}} = (\pi_{\mathcal{D}}Q_{\mathcal{D}})R_{\mathcal{D}} = 0$ since $\pi_{\mathcal{D}}H_{\pi_{\mathcal{D}}} = \pi_{\mathcal{D}}$ from Lemma 3.10 and $\pi_{\mathcal{D}}Q_{\mathcal{D}} = 0$ by assumption.

The next subsection specifies sufficient conditions for a converging smoother to provide improved solutions at each level.

### 4.2. Convergence of the smoothers.

By definition at the finest level in Algorithm 1 and by construction at the coarser levels in Algorithm 2, the matrix $\tilde{Q}_{\mathcal{D}}$ is an irreducible CTMC when $x'_{\mathcal{D}} > 0$ (see Lemma 4.1). Now, consider the nontransposed homogeneous singular linear system in the next definition (cf. (1.1)).

Definition 4.4. [illegible] $\mathcal{D}$ [illegible]

$$\tilde{\pi}_{\mathcal{D}}\tilde{Q}_{\mathcal{D}} = 0 \quad \text{[illegible]} \quad \tilde{\pi}_{\mathcal{D}}e = 1,$$

[illegible] $\tilde{\pi}_{\mathcal{D}} > 0$ [illegible] $\tilde{Q}_{\mathcal{D}}$

Proposition 4.5. [illegible] $\mathcal{D}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $\tilde{\pi}_{\mathcal{D}} = \pi$ [illegible] $\tilde{Q}_{\mathcal{D}} = Q$

Now, consider the splitting of $\tilde{Q}_{\mathcal{D}}$ in the next definition.

Definition 4.6. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible]

$$\tilde{Q}_{\mathcal{D}} = D_{\mathcal{D}} - U_{\mathcal{D}} - L_{\mathcal{D}} = M_{\mathcal{D}} - N_{\mathcal{D}},$$

[illegible] $D_{\mathcal{D}}$ $U_{\mathcal{D}}$ [illegible] $L_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $M_{\mathcal{D}}$ [illegible] $M_{\mathcal{D}}^{-1}$ [illegible]

Proposition 4.7. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $D_{\mathcal{D}}$ $U_{\mathcal{D}}$ [illegible] $L_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}}) \neq 0$ [illegible] $s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}$ [illegible] $D_{\mathcal{D}}^{-1}$ [illegible] $(D_{\mathcal{D}} - U_{\mathcal{D}})^{-1}$ [illegible]

The next definition involving the iteration matrices of the *POWER*, *JOR*, and *SOR* smoothers follows from [29, Chap. 3].

Proposition 4.8. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible]

$$T_{\mathcal{D}} = N_{\mathcal{D}}M_{\mathcal{D}}^{-1}$$

[illegible]

$$x_{\mathcal{D}}^{(m+1)} = x_{\mathcal{D}}^{(m)}T_{\mathcal{D}} \quad \text{[illegible]} \quad m = 0, 1, \ldots.$$

[illegible]

$$M_{\mathcal{D}}^{POWER} = -\alpha_{\mathcal{D}} I_{\mathcal{D}}, \quad N_{\mathcal{D}}^{POWER} = -\alpha_{\mathcal{D}}(I_{\mathcal{D}} + \tilde{Q}_{\mathcal{D}}/\alpha_{\mathcal{D}}),$$
$$M_{\mathcal{D}}^{JOR} = D_{\mathcal{D}}/\omega, \quad N_{\mathcal{D}}^{JOR} = (1-\omega)D_{\mathcal{D}}/\omega + L_{\mathcal{D}} + U_{\mathcal{D}},$$
$$M_{\mathcal{D}}^{SOR} = D_{\mathcal{D}}/\omega - U_{\mathcal{D}}, \quad N_{\mathcal{D}}^{SOR} = (1-\omega)D_{\mathcal{D}}/\omega + L_{\mathcal{D}},$$

[illegible] $\alpha_{\mathcal{D}} \in [\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0,2)$ [illegible] $\omega = 1$ [illegible]

$$T_{\mathcal{D}}^{POWER} = I_{\mathcal{D}} + \tilde{Q}_{\mathcal{D}}/\alpha_{\mathcal{D}},$$
$$T_{\mathcal{D}}^{JOR} = (1-\omega)I_{\mathcal{D}} + \omega(L_{\mathcal{D}} + U_{\mathcal{D}})D_{\mathcal{D}}^{-1},$$
$$T_{\mathcal{D}}^{SOR} = ((1-\omega)D_{\mathcal{D}}/\omega + L_{\mathcal{D}})(D_{\mathcal{D}}/\omega - U_{\mathcal{D}})^{-1}.$$

Since $\tilde{Q}_{\mathcal{D}}$ is the generator matrix of an irreducible CTMC, the relation $\tilde{\pi}_{\mathcal{D}} T_{\mathcal{D}}^{S} = \tilde{\pi}_{\mathcal{D}}$ holds for $S \in \{POWER, SOR, JOR\}$ [29].

Before we state another lemma, we recall the definitions of primitivity and M-matrix from [29, pp. 352, 170] and remark that detailed information concerning M-matrices may be found in [4].

DEFINITION 4.9. [illegible] $\sigma(A)$ [illegible] $A$ [illegible] $\rho(A)$ [illegible] $A$ [illegible] $\rho(A) = \{\max |\lambda| \mid \lambda \in \sigma(A)\}$ [illegible] $B$ [illegible] $\rho(B)$

DEFINITION 4.10. [illegible] $A$ [illegible] $A = \beta I - B$ [illegible] $\beta > 0$ [illegible] $B \geq 0$ [illegible] $\beta \geq \rho(B)$ [illegible]

Hence, the negated CTMC $-\tilde{Q}_{\mathcal{D}}$ is a singular M-matrix. The next proposition follows from [23, p. 640] and [29, p. 118].

PROPOSITION 4.11. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $e\tilde{\pi}_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] 1

COROLLARY 4.12. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $|\mathcal{S}_{\mathcal{D}}| = 1$ $\tilde{Q}_{\mathcal{D}} = 0$ [illegible] $\tilde{\pi}_{\mathcal{D}} = 1$

For HMMs, Corollary 4.12 applies at the coarsest level when the HLM has one macrostate.

Now we are in a position to state and prove a lemma, which is essential in characterizing the convergence of the three smoothers.

LEMMA 4.13. [illegible] $S \in \{POWER, JOR, SOR\}$ [illegible] $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0,1)$ [illegible] $T_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] 1 [illegible] $T_{\mathcal{D}} = W_{\mathcal{D}} B_{\mathcal{D}} W_{\mathcal{D}}^{-1}$ [illegible] $B_{\mathcal{D}}$ [illegible] $W_{\mathcal{D}}$ [illegible] $T_{\mathcal{D}}$ [illegible] $\lim_{m\to\infty} T_{\mathcal{D}}^{m} = (W_{\mathcal{D}} e)\tilde{\pi}_{\mathcal{D}}/(\tilde{\pi}_{\mathcal{D}} W_{\mathcal{D}} e) > 0$ [illegible] 1 [illegible] $POWER$ [illegible] $W_{\mathcal{D}} = I_{\mathcal{D}}$ [illegible] $T_{\mathcal{D}}$ [illegible] $\lim_{m\to\infty} T_{\mathcal{D}}^{m} = e\tilde{\pi}_{\mathcal{D}} > 0$

[illegible] The proof follows from Theorem 17 of [29]. □

Using Lemma 4.13, the next proposition expresses the pre- and postsmoothings at level $\mathcal{D}$ concisely.

PROPOSITION 4.14. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $x_{\mathcal{D}} > 0$ [illegible] $\nu > 0$ [illegible] $\mathcal{D}$ [illegible]

[illegible] $S$ [illegible] 4.13 [illegible]

$$x'_{\mathcal{D}} = x_{\mathcal{D}} T^{\nu}_{\mathcal{D}} > 0.$$

The next proposition follows from Theorem 4.4 in [28, pp. 45–46] and is introduced to aid the characterization of the nonasymptotic convergence behavior of smoothings.

PROPOSITION 4.15. [illegible] $A_{\mathcal{D}} \in \mathbb{R}^{|\mathcal{S}_{\mathcal{D}}| \times |\mathcal{S}_{\mathcal{D}}|}$ [illegible] $A_{\mathcal{D}}^{-1}$ [illegible]

$$\|w\|_{A_{\mathcal{D}}} = \|wA_{\mathcal{D}}\|_1 \quad \text{[illegible]} \quad w \in \mathbb{R}^{1 \times |\mathcal{S}_{\mathcal{D}}|}$$

[illegible][2]

The next theorem characterizes the [illegible] behavior of the smoothings through a lemma for positive stochastic matrices based on the discussion in [2, pp. 270–271] and proved in [13, appendix], and two results on nonnegative irreducible matrices similar to positive matrices [5, pp. 371 and 375]. We remark that a similar theorem may be stated for the initial approximation $y_{\mathcal{D}}$.

THEOREM 4.16. [illegible] $x_{\mathcal{D}}^{(0)} = x_{\mathcal{D}} > 0$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $S \in \{POWER, JOR, SOR\}$ [illegible] $T_{\mathcal{D}}$ [illegible] $x_{\mathcal{D}}^T \notin \text{Range}(I_{\mathcal{D}} - T_{\mathcal{D}}^T)$ [illegible] $T_{\mathcal{D}}^{\nu_1}$ [illegible]

(i) $T_{\mathcal{D}}^{\nu_1}$ [illegible]

(ii) $T_{\mathcal{D}}^{\nu_1}$ [illegible] $i_{\mathcal{D}}$ [illegible] $j_{\mathcal{D}}$

(iii) $T_{\mathcal{D}}^{\nu_1}$ [illegible] $(i_{\mathcal{D}}, j_{\mathcal{D}})$

(a) [illegible] $i_{\mathcal{D}}$ [illegible] $e_{i_{\mathcal{D}}}^T T_{\mathcal{D}}^{\nu_1} e_{i_{\mathcal{D}}} > e_{j_{\mathcal{D}}}^T T_{\mathcal{D}}^{\nu_1} e_{j_{\mathcal{D}}}$ [illegible]

(b) [illegible] $j_{\mathcal{D}}$ [illegible] $e_{i_{\mathcal{D}}}^T T_{\mathcal{D}}^{\nu_1} e_{i_{\mathcal{D}}} < e_{j_{\mathcal{D}}}^T T_{\mathcal{D}}^{\nu_1} e_{j_{\mathcal{D}}}$

[illegible]

$$\|c_{\mathcal{D}} x'_{\mathcal{D}} - \tilde{\pi}_{\mathcal{D}}\|_{A_{\mathcal{D}}} \leq \left(1 - \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} g_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}})\right) \|c_{\mathcal{D}} x_{\mathcal{D}} - \tilde{\pi}_{\mathcal{D}}\|_{A_{\mathcal{D}}},$$

[illegible] $x'_{\mathcal{D}} = x_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1}$ $G_{\mathcal{D}}$ [illegible] $G_{\mathcal{D}} = A_{\mathcal{D}}^{-1} T_{\mathcal{D}}^{\nu_1} A_{\mathcal{D}}$ [illegible] $A_{\mathcal{D}} \geq 0$ [illegible] $0 < \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} g_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}}) \leq 1/|\mathcal{S}_{\mathcal{D}}|$ $\tilde{\pi}_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $c_{\mathcal{D}} = (\tilde{\pi}_{\mathcal{D}} A_{\mathcal{D}} e)/(x_{\mathcal{D}} A_{\mathcal{D}} e)$

[illegible] From Corollary 3 and Theorem 4 in [5], if $T_{\mathcal{D}}^{\nu_1}$ is nonnegative, is irreducible, and satisfies either of the conditions (ii) or (iii), then it is similar to a positive matrix; that is, $X_{\mathcal{D}}^{-1} T_{\mathcal{D}}^{\nu_1} X_{\mathcal{D}} = H_{\mathcal{D}} > 0$ for some $(|\mathcal{S}_{\mathcal{D}}| \times |\mathcal{S}_{\mathcal{D}}|)$ nonnegative matrix $X_{\mathcal{D}}$. Condition (i) is a special case for which $X_{\mathcal{D}} = I_{\mathcal{D}}$. Since these imply $\sigma(H_{\mathcal{D}}) = \sigma(T_{\mathcal{D}}^{\nu_1})$ and we have $\rho(T_{\mathcal{D}}^{\nu_1}) = 1$ from Lemma 4.13, $H_{\mathcal{D}} > 0$ must be similar to a positive stochastic matrix $G_{\mathcal{D}}$ as in $Y_{\mathcal{D}}^{-1} H_{\mathcal{D}} Y_{\mathcal{D}} = G_{\mathcal{D}} > 0$, where $Y_{\mathcal{D}}$ is a nonnegative diagonal matrix having the positive right eigenvector of $H_{\mathcal{D}}$ along its diagonal. Now, let $A_{\mathcal{D}} = X_{\mathcal{D}} Y_{\mathcal{D}}$ to obtain $T_{\mathcal{D}}^{\nu_1} = A_{\mathcal{D}} G_{\mathcal{D}} A_{\mathcal{D}}^{-1}$, where $A_{\mathcal{D}} \geq 0$, $G_{\mathcal{D}} > 0$, and $G_{\mathcal{D}} e = e$.

For a sequence of converging approximations, one needs to ensure for the initial approximation that $x_{\mathcal{D}}^T \notin \text{Range}(I_{\mathcal{D}} - T_{\mathcal{D}}^T)$ [3, pp. 26–28]; otherwise, there will be no improvement. Furthermore, since $\tilde{\pi}_{\mathcal{D}}$ is the unique positive fixed point of $T_{\mathcal{D}}^{\nu_1}$ such that $\tilde{\pi}_{\mathcal{D}} e = 1$, the unique positive fixed point of $G_{\mathcal{D}}$ with unit 1-norm must be $\psi_{\mathcal{D}} = (\tilde{\pi}_{\mathcal{D}} A_{\mathcal{D}})/(\tilde{\pi}_{\mathcal{D}} A_{\mathcal{D}} e)$. Now, rewrite $x'_{\mathcal{D}} = x_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1}$ using $T_{\mathcal{D}}^{\nu_1} = A_{\mathcal{D}} G_{\mathcal{D}} A_{\mathcal{D}}^{-1}$

---

[2] This norm should not be confused with the elliptical norm [23, p. 288] defined as $\|w\|_{A_{\mathcal{D}}} = \|wA_{\mathcal{D}}\|_2$.

to obtain $x'_{\mathcal{D}}A_{\mathcal{D}} = x_{\mathcal{D}}A_{\mathcal{D}}(G_{\mathcal{D}})$. Since $x_{\mathcal{D}} > 0$, $A_{\mathcal{D}} \geq 0$, and $A_{\mathcal{D}}$ has full rank, we have $x'_{\mathcal{D}} > 0$. Furthermore, note that $x'_{\mathcal{D}}A_{\mathcal{D}}e = x_{\mathcal{D}}A_{\mathcal{D}}(G_{\mathcal{D}}e) = x_{\mathcal{D}}A_{\mathcal{D}}e$. Letting $\overline{x}'_{\mathcal{D}} = (x'_{\mathcal{D}}A_{\mathcal{D}})/(x_{\mathcal{D}}A_{\mathcal{D}}e)$ and $\overline{x}_{\mathcal{D}} = (x_{\mathcal{D}}A_{\mathcal{D}})/(x_{\mathcal{D}}A_{\mathcal{D}}e)$, we have from Lemma A.1 in [13, appendix]

$$\|\overline{x}'_{\mathcal{D}} - \psi_{\mathcal{D}}\|_1 \leq \left(1 - \min_{i_{\mathcal{D}},j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} g_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}})\right) \|\overline{x}_{\mathcal{D}} - \psi_{\mathcal{D}}\|_1.$$

The result follows by taking each of $(\overline{x}'_{\mathcal{D}} - \psi_{\mathcal{D}})$ and $(\overline{x}_{\mathcal{D}} - \psi_{\mathcal{D}})$ into $A_{\mathcal{D}}$ parentheses, multiplying both sides of the inequality by $\tilde{\pi}_{\mathcal{D}}A_{\mathcal{D}}e$, letting $c_{\mathcal{D}} = (\tilde{\pi}_{\mathcal{D}}A_{\mathcal{D}}e)/(x_{\mathcal{D}}A_{\mathcal{D}}e)$, and using Proposition 4.15. □

Theorem 4.16 indicates that the [illegible] solution vector, $c_{\mathcal{D}}x_{\mathcal{D}}$, improves with $\nu_1$ presmoothings if $T_{\mathcal{D}}^{\nu_1}$ is positive or has a(n almost) positive row or column. Now, observe that the ordering of grids suggested by $O \in \{FIXED, CIRCULAR, DYNAMIC\}$ has no effect on the assumptions of Theorem 4.16. Note also from Lemma 4.13 that as $\nu_1$ increases, $T_{\mathcal{D}}^{\nu_1}$ converges to a positive rank 1 matrix. Hence, there is a value of $\nu_1 > 0$ for which the assumptions of Theorem 4.16 hold. We remark that $\tilde{Q}_{\mathcal{D}}$ is almost always sparse, and the iteration matrices associated with the *POWER* and *JOR* smoothers have the same off-diagonal nonzero structure as that of $\tilde{Q}_{\mathcal{D}}$. Hence, compared to *POWER* and *JOR*, the *SOR* smoother has a higher chance of satisfying the conditions of Theorem 4.16 for a smaller value of $\nu_1$, since its iteration matrix is likely to have a larger number of nonzeros, as suggested in the proof of Lemma 4.17 in [13]. Similar arguments are valid for postsmoothings. These results can be perceived as an extension of the local convergence result available in [22, sec. 2] to include the *JOR* and *SOR* smoothers and another sufficient condition (i.e., Theorem 4.16(iii)). In summary, the smoothings can always be enforced to yield improved positive approximations at each level.

**4.3. Convergence of the ML solver.** Using the results in the previous subsections, we show that under certain conditions the devised class of ML methods provide converging iterations for different choices of the cycle parameter $C \in \{V, W, F\}$.

First, we define the ML iteration matrix at level $\mathcal{D}$ in Algorithms 1 and 2 using Propositions 3.5, 3.7, 4.11, and 4.14. Note that when there are only two levels, the W- and F-cycles are not defined, and the V-cycle yields a two-level IAD solver. In order not to complicate the notation further, we refrain from introducing an index for the cycle number to the matrices and vectors at this point.

DEFINITION 4.17. [illegible] $T_{\mathcal{D}}^{ML}$ [illegible] $\mathcal{D}$ [illegible] $x_{\mathcal{D}} > 0$ [illegible] $y'_{\mathcal{D}} > 0$ [illegible] $S \in \{POWER, JOR, SOR\}$ [illegible] $T_{\mathcal{D}}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0, 1)$ [illegible] $R_{\mathcal{D}}$ [illegible] $P_{x'_{\mathcal{D}}}$ [illegible] $T_{\mathcal{C}}^{ML}$ [illegible] $T_{\mathcal{B}}^{ML}$ [illegible] $\mathcal{C}$ [illegible] $\mathcal{B}$ [illegible]

$$y'_{\mathcal{D}} = x_{\mathcal{D}}T_{\mathcal{D}}^{ML},$$

[illegible]

$$T_{\mathcal{D}}^{ML} = \begin{cases} T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}} T_{\mathcal{C}}^{ML} P_{x'_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} & [illegible]\ C = V, \\ T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}} (T_{\mathcal{C}}^{ML})^2 P_{x'_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} & [illegible]\ C = W, \\ T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}} T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} P_{x'_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} & [illegible]\ C = F, \end{cases}$$

$$T_{\mathcal{C}}^{ML'} = T_{\mathcal{C}}^{\nu_1} R_{\mathcal{C}} T_{\mathcal{B}}^{ML'} P_{x'_{\mathcal{C}}} T_{\mathcal{C}}^{\nu_2}, \quad x'_{\mathcal{D}} = x_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1},$$

[illegible] $\tilde{Q}_{\mathcal{C}}$ [illegible] $T_{\mathcal{C}}^{ML} = T_{\mathcal{C}}^{ML'} = (ey'_{\mathcal{C}})/(x_{\mathcal{C}}e) > 0$ [illegible] $y_{\mathcal{C}} = \tilde{\pi}_{\mathcal{C}}$

COROLLARY 4.18. [illegible] $POWER$ [illegible] $x_{\mathcal{D}} > 0$ [illegible] $x_{\mathcal{D}}e = 1$ [illegible] $T_{\mathcal{D}}^{ML}$ [illegible] $C \in \{V, W, F\}$ [illegible] 1

[illegible] For the $POWER$ smoother, at the coarsest level $\mathcal{C}$ we have $T_{\mathcal{C}}^{ML} = T_{\mathcal{C}}^{ML'} = e\tilde{\pi}_{\mathcal{C}}$ from Definition 4.17 when $x_{\mathcal{D}}e = 1$, implying a positive stochastic matrix, which has a spectral radius and an eigenvalue value of 1. This forms the base case. Now, let us assume that the result is true for all levels from the coarsest up to an arbitrary level $\mathcal{C}$; this is the inductive hypothesis. We show that the result must be true for the next finer level $\mathcal{D}$. Noting that $R_{\mathcal{D}}e = e$ from Proposition 3.5, $(T_{\mathcal{C}}^{ML})e = e$ from the inductive hypothesis, $P_{x'_{\mathcal{D}}}e = e$ from Proposition 3.7, and $T_{\mathcal{D}}e = e$ from Lemma 4.13, we have $T_{\mathcal{D}}^{ML}e = T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}} T_{\mathcal{C}}^{ML} P_{x'_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} e = T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}} T_{\mathcal{C}}^{ML} (P_{x'_{\mathcal{D}}} e) = T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}} (T_{\mathcal{C}}^{ML} e) = T_{\mathcal{D}}^{\nu_1} (R_{\mathcal{D}} e) = T_{\mathcal{D}}^{\nu_1} e = e$ for the V-cycle. The result follows similarly for W- and F-cycles. □

The interpretation of $T_{\mathcal{D}}^{ML}$ for V- and W-cycles is as follows. If the recursive call(s) to level $\mathcal{C}$ are turned off, then only $(\nu_1 + \nu_2)$ iterations are performed on $x_{\mathcal{D}}$ with the smoother $S$. Otherwise, the smoothed solution vector is restricted to level $\mathcal{C}$ (i.e., $x_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1}$ is the smoothed solution vector and $x_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1} R_{\mathcal{D}}$ is the restricted solution vector), the restricted solution vector is improved respectively one or two times with the iteration matrix $T_{\mathcal{C}}^{ML}$, and the improved solution vector is projected back to level $\mathcal{D}$ and smoothed. The interpretation of $T_{\mathcal{D}}^{ML}$ for an F-cycle is similar to that for V- and W-cycles with the difference that the restricted solution vector is improved with the iteration matrix $T_{\mathcal{D}}^{ML}$ once followed by the iteration matrix of the V-cycle. This is exactly what is meant with a W-cycle followed by a V-cycle at each level.

The next lemma follows from Lemma 4.1, Lemma 4.13, and Definition 4.17.

LEMMA 4.19. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $x_{\mathcal{D}} > 0$ [illegible] $S \in \{POWER, JOR, SOR\}$ [illegible] $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0, 1)$ [illegible] $T_{\mathcal{D}}^{ML}$ [illegible] $C \in \{V, W, F\}$ [illegible]

[illegible] The proof is by induction. At the coarsest level $\mathcal{C}$, we have $T_{\mathcal{C}}^{ML} = T_{\mathcal{C}}^{ML'} > 0$ from Definition 4.17. This is the base case and implies $(T_{\mathcal{C}}^{ML})^2 = T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} > 0$. Now, let us assume that the statement is true for all levels from the coarsest up to an arbitrary level $\mathcal{C}$. This is the inductive hypothesis. We show that the statement must be true for the next finer level $\mathcal{D}$. Since $P_{x'_{\mathcal{D}}} \geq 0$ and each column of $P_{x'_{\mathcal{D}}}$ has one nonzero element from Proposition 3.7, the $(|\mathcal{S}_{\mathcal{C}}| \times |\mathcal{S}_{\mathcal{D}}|)$ matrices $T_{\mathcal{C}}^{ML} P_{x'_{\mathcal{D}}}$, $(T_{\mathcal{C}}^{ML})^2 P_{x'_{\mathcal{D}}}$, and $T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} P_{x'_{\mathcal{D}}}$ are positive. Furthermore, since $R_{\mathcal{D}} \geq 0$ and each row of $R_{\mathcal{D}}$ has one nonzero element from Proposition 3.5, the $(|\mathcal{S}_{\mathcal{D}}| \times |\mathcal{S}_{\mathcal{D}}|)$ matrices $R_{\mathcal{D}} T_{\mathcal{C}}^{ML} P_{x'_{\mathcal{D}}}$, $R_{\mathcal{D}} (T_{\mathcal{C}}^{ML})^2 P_{x'_{\mathcal{D}}}$, and $R_{\mathcal{D}} T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} P_{x'_{\mathcal{D}}}$ are also positive. Then the result follows from Lemma 4.13 by the fact that the iteration matrix associated with the smoother is nonnegative and irreducible, implying at least one nonzero in each row and column of $T_{\mathcal{D}}$ which pre- and postmultiplies the positive matrices $R_{\mathcal{D}} T_{\mathcal{C}}^{ML} P_{x'_{\mathcal{D}}}$, $R_{\mathcal{D}} (T_{\mathcal{C}}^{ML})^2 P_{x'_{\mathcal{D}}}$, and $R_{\mathcal{D}} T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} P_{x'_{\mathcal{D}}}$. □

The next result follows from Lemma 4.19 in that the positivity of $T_{\mathcal{D}}^{ML}$ implies its irreducibility and a positive diagonal, and hence its primitivity [4, p. 47].

COROLLARY 4.20. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $x_{\mathcal{D}} > 0$ [illegible] $S \in$

$\{POWER, JOR, SOR\}$ [illegible] $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0,1)$ [illegible] $T_{\mathcal{D}}^{ML}$ [illegible] $C \in \{V, W, F\}$ [illegible]

The next lemma shows that the steady state vector, $\pi_{\mathcal{D}}$, of the exactly aggregated grid, $Q_{\mathcal{D}}$, is the unique, positive, unit 1-norm fixed point of the ML iteration matrix, $T_{\mathcal{D}}^{ML}$, at level $\mathcal{D}$ upon convergence.

LEMMA 4.21. [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $Q_{\mathcal{D}}$ $x_{\mathcal{D}} = \pi_{\mathcal{D}}$ [illegible] $S \in \{POWER, JOR, SOR\}$ [illegible] $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0,1)$ [illegible] $T_{\mathcal{D}}^{ML}$ [illegible] $C \in \{V, W, F\}$ [illegible] $\pi_{\mathcal{D}}$ [illegible] $\pi_{\mathcal{D}} T_{\mathcal{D}}^{ML} = \pi_{\mathcal{D}}$ [illegible] $\pi_{\mathcal{D}} e = 1$ [illegible] $\rho(T_{\mathcal{D}}^{ML}) = 1$ [illegible] $y_{\mathcal{D}}' = \pi_{\mathcal{D}}$

[illegible] The proof is by induction. At the coarsest level $\mathcal{C}$, we have $\tilde{Q}_{\mathcal{C}} = Q_{\mathcal{C}}$ and $x_{\mathcal{C}} = \pi_{\mathcal{C}} > 0$, implying $T_{\mathcal{C}}^{ML} = T_{\mathcal{C}}^{ML'} = e\pi_{\mathcal{C}} > 0$ from Definition 4.17. This positive matrix is stochastic and has the unique positive fixed point $\pi_{\mathcal{C}}$ such that $\pi_{\mathcal{C}} e = 1$. Furthermore, it has a spectral radius of 1 and $y_{\mathcal{C}}' = x_{\mathcal{C}} T_{\mathcal{C}}^{ML} = \pi_{\mathcal{C}}(e\pi_{\mathcal{C}}) = (\pi_{\mathcal{C}} e)\pi_{\mathcal{C}} = \pi_{\mathcal{C}}$. This is the base case and yields $(T_{\mathcal{C}}^{ML})^2 = T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} = (e\pi_{\mathcal{C}})(e\pi_{\mathcal{C}}) = e(\pi_{\mathcal{C}} e)\pi_{\mathcal{C}} = e\pi_{\mathcal{C}} > 0$. Now, let us assume that the statement is true for all levels from the coarsest up to an arbitrary level $\mathcal{C}$. This is the inductive hypothesis. We show that the statement must be true for the next finer level $\mathcal{D}$.

Since $x_{\mathcal{D}} = \pi_{\mathcal{D}} > 0$ is the fixed point of $T_{\mathcal{D}}$, $\pi_{\mathcal{D}} R_{\mathcal{D}} = \pi_{\mathcal{C}}$ from Definition 3.4, $\pi_{\mathcal{C}} T_{\mathcal{C}}^{ML} = \pi_{\mathcal{C}}$ by the inductive hypothesis, and $\pi_{\mathcal{C}} P_{\pi_{\mathcal{D}}} = \pi_{\mathcal{D}}$ from Definition 3.6, the result follows from Definition 4.17 for the V-cycle as $y_{\mathcal{D}}' = \pi_{\mathcal{D}} T_{ML}^{D} = (\pi_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1}) R_{\mathcal{D}} T_{\mathcal{C}}^{ML} P_{\pi_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} = (\pi_{\mathcal{D}} R_{\mathcal{D}}) T_{\mathcal{C}}^{ML} P_{\pi_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} = (\pi_{\mathcal{C}} T_{\mathcal{C}}^{ML}) P_{\pi_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} = (\pi_{\mathcal{C}} P_{\pi_{\mathcal{D}}}) T_{\mathcal{D}}^{\nu_2} = \pi_{\mathcal{D}} T_{\mathcal{D}}^{\nu_2} = \pi_{\mathcal{D}}$. The result follows similarly for W- and F-cycles after interchanging $T_{\mathcal{C}}^{ML}$ respectively with $(T_{\mathcal{C}}^{ML})^2$ and $T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'}$. The uniqueness and positivity of the fixed point of $T_{\mathcal{D}}^{ML}$ follows from Lemma 4.19 by the fact that $T_{\mathcal{D}}^{ML}$ is positive [23, p. 666]. Clearly the spectral radius of $T_{\mathcal{D}}^{ML}$ is 1. □

The next theorem characterizes the [illegible] behavior of the ML solver with the initial approximation $x_{\mathcal{D}}$ by defining a unique, positive, unit 1-norm fixed point for the particular cycle.

THEOREM 4.22. [illegible] $T_{\mathcal{D}}^{ML}$ [illegible] $\mathcal{D}$ [illegible] $x_{\mathcal{D}} > 0$ [illegible] $x_{\mathcal{D}}^T \notin \mathrm{Range}(I_{\mathcal{D}} - T_{\mathcal{D}}^T)$ [illegible] $y_{\mathcal{D}}' > 0$ [illegible] $S \in \{POWER, JOR, SOR\}$ [illegible] $\tilde{Q}_{\mathcal{D}}$ [illegible] $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$ [illegible] $\omega \in (0,1)$ [illegible] $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$ [illegible] 1 [illegible] $\phi_{\mathcal{D}}$ [illegible] $\phi_{\mathcal{D}}(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})) = \phi_{\mathcal{D}}$ [illegible] $\phi_{\mathcal{D}} e = 1$ [illegible] $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = Z_{\mathcal{D}} H_{\mathcal{D}} Z_{\mathcal{D}}^{-1}$ [illegible] $H_{\mathcal{D}}$ [illegible] $Z_{\mathcal{D}}$ [illegible] $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$ [illegible] $H_{\mathcal{D}}$ [illegible] $\psi_{\mathcal{D}} = (\phi_{\mathcal{D}} Z_{\mathcal{D}})/(\phi_{\mathcal{D}} Z_{\mathcal{D}} e)$ [illegible] $\psi_{\mathcal{D}} H_{\mathcal{D}} = \psi_{\mathcal{D}}$ [illegible] $\psi_{\mathcal{D}} e = 1$ [illegible]

$$\|(b_{\mathcal{D}}/\rho(T_{\mathcal{D}}^{ML}))y_{\mathcal{D}}' - \phi_{\mathcal{D}}\|_{Z_{\mathcal{D}}} \le \left(1 - \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} h_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}})\right) \|b_{\mathcal{D}} x_{\mathcal{D}} - \phi_{\mathcal{D}}\|_{Z_{\mathcal{D}}},$$

[illegible] $b_{\mathcal{D}} = (\phi_{\mathcal{D}} Z_{\mathcal{D}} e)/(x_{\mathcal{D}} Z_{\mathcal{D}} e)$ [illegible] $0 < \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} h_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}}) \le 1/|\mathcal{S}_{\mathcal{D}}|$ [illegible] $\|(b_{\mathcal{D}}/\rho(T_{\mathcal{D}}^{ML}))y_{\mathcal{D}}' - \phi_{\mathcal{D}}\|_{Z_{\mathcal{D}}} = 0$ [illegible] $POWER$ [illegible] $Z_{\mathcal{D}} = I_{\mathcal{D}}$ $H_{\mathcal{D}} = T_{\mathcal{D}}^{ML}$ $\rho(T_{\mathcal{D}}^{ML}) = 1$ $\psi_{\mathcal{D}} = \phi_{\mathcal{D}}$ [illegible] $b_{\mathcal{D}} = 1$

[illegible] Recall from Lemma 4.19 that $T_{\mathcal{D}}^{ML} > 0$. Since $\rho(T_{\mathcal{D}}^{ML}) > 0$ for $T_{\mathcal{D}}^{ML} \neq 0$, the matrix $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$ is also positive, satisfies $\sigma(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})) = \{\lambda/\rho(T_{\mathcal{D}}^{ML}) \mid \lambda \in \sigma(T_{\mathcal{D}}^{ML})\}$, and therefore has a spectral radius of 1. The uniqueness and positivity of the fixed point $\phi_{\mathcal{D}}$ follow from Corollary 4.20. The row vector $\phi_{\mathcal{D}} > 0$ is assumed to be normalized so as to have unit 1-norm (i.e., $\phi_{\mathcal{D}} e = 1$).

To prove the second part, recall Corollary 4.20 and the result in [4, p. 49], which is also used in the proof of Lemma 4.13. These imply that $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$ must have a positive right eigenvector $\zeta_{\mathcal{D}}$ for which

$$T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = Z_{\mathcal{D}} H_{\mathcal{D}} Z_{\mathcal{D}}^{-1},$$

where $Z_{\mathcal{D}} = \mathrm{diag}(\zeta_{\mathcal{D}})$, $H_{\mathcal{D}} > 0$, and $H_{\mathcal{D}} e = e$. In other words,

$$H_{\mathcal{D}} = Z_{\mathcal{D}}^{-1}(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}))Z_{\mathcal{D}}$$

is a stochastic matrix similar to $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$, and its positivity follows from $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) > 0$ and $\zeta_{\mathcal{D}} > 0$. Note that it does not matter whether $\zeta_{\mathcal{D}}$ is normalized or not, since $H_{\mathcal{D}}$ is defined in terms of $Z_{\mathcal{D}}$ and $Z_{\mathcal{D}}^{-1}$. The uniqueness and positivity of the fixed point $\psi_{\mathcal{D}}$ follows from $H_{\mathcal{D}} > 0$. The row vector $\phi_{\mathcal{D}} > 0$ is assumed to be normalized so as to have unit 1-norm (i.e., $\phi_{\mathcal{D}} e = 1$), and it is given by $\psi_{\mathcal{D}} = \phi_{\mathcal{D}} Z_{\mathcal{D}}$ since $H_{\mathcal{D}}$ and $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$ are related by a similarity transformation, where the transformation matrix is $Z_{\mathcal{D}}$.

To prove the last part, rewrite

$$y_{\mathcal{D}}^{'} = \rho(T_{\mathcal{D}}^{ML}) x_{\mathcal{D}} (T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}))$$

using $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = Z_{\mathcal{D}} H_{\mathcal{D}} Z_{\mathcal{D}}^{-1} > 0$ as

$$(y_{\mathcal{D}}^{'} Z_{\mathcal{D}})/(\rho(T_{\mathcal{D}}^{ML}) x_{\mathcal{D}} Z_{\mathcal{D}} e) = (x_{\mathcal{D}} Z_{\mathcal{D}}) H_{\mathcal{D}}/(x_{\mathcal{D}} Z_{\mathcal{D}} e),$$

which is equivalent to $\overline{y}_{\mathcal{D}}^{'} = \overline{x}_{\mathcal{D}} H_{\mathcal{D}}$. Since $x_{\mathcal{D}} > 0$, $y_{\mathcal{D}}^{'} > 0$, $\rho(T_{\mathcal{D}}^{ML}) > 0$, and $\zeta_{\mathcal{D}} > 0$, we have $\overline{x}_{\mathcal{D}} = (x_{\mathcal{D}} Z_{\mathcal{D}})/(x_{\mathcal{D}} Z_{\mathcal{D}} e) > 0$, implying $\overline{x}_{\mathcal{D}} e = 1$, and $\overline{y}_{\mathcal{D}}^{'} = (y_{\mathcal{D}}^{'} Z_{\mathcal{D}})/(\rho(T_{\mathcal{D}}^{ML}) x_{\mathcal{D}} Z_{\mathcal{D}} e) > 0$. Furthermore, since $H_{\mathcal{D}} > 0$, $H_{\mathcal{D}} e = e$, and $\overline{x}_{\mathcal{D}} e = 1$, we obtain $\overline{y}_{\mathcal{D}}^{'} e = 1$. Then, from Lemma A.1 in [13, appendix] we have

$$\|\overline{y}_{\mathcal{D}}^{'} - \psi_{\mathcal{D}}\|_1 \leq \left(1 - \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} h_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}})\right) \|\overline{x}_{\mathcal{D}} - \psi_{\mathcal{D}}\|_1.$$

The result follows by taking each of $(\overline{y}_{\mathcal{D}}^{'} - \psi_{\mathcal{D}})$ and $(\overline{x}_{\mathcal{D}} - \psi_{\mathcal{D}})$ into $Z_{\mathcal{D}}$ parentheses, multiplying both sides of the inequality by $\phi_{\mathcal{D}} Z_{\mathcal{D}} e$, letting $b_{\mathcal{D}} = (\phi_{\mathcal{D}} Z_{\mathcal{D}} e)/(x_{\mathcal{D}} Z_{\mathcal{D}} e)$, and using Proposition 4.15. The part for the coarsest level follows from Definition 4.17 by the fact that $T_{\mathcal{D}}^{ML} = (e y_{\mathcal{D}}^{'})/(x_{\mathcal{D}} e)$ and $\rho(T_{\mathcal{D}}^{ML}) = 1/(x_{\mathcal{D}} e)$, implying $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = H_{\mathcal{D}} = e y_{\mathcal{D}}^{'}$ and $Z_{\mathcal{D}} = I_{\mathcal{D}}$. For the *POWER* smoother, Corollary 4.18 implies $Z_{\mathcal{D}} = \mathrm{diag}(e) = I_{\mathcal{D}}$, and therefore, the respective results. ☐

The ML iteration matrix, $T_{\mathcal{D}}^{ML}$, changes at each cycle due to the dependence of $P_{x_{\mathcal{D}}^{'}}$ on $x_{\mathcal{D}}^{'}$, and therefore the ML iteration is nonstationary. At the end of each ML iteration, the solution vector at the finest level $\mathcal{D}$, $y_{\mathcal{D}}^{'}$, is normalized to be unit 1-norm and then assigned to $x_{\mathcal{D}}$ so as to start the next ML iteration. As long as $x_{\mathcal{D}}^{'} \neq \pi_{\mathcal{D}}$, the aggregated CTMC $\tilde{Q}_{\mathcal{C}}$ at the next coarser level can be only approximative. Theorem 4.22 indicates that the [illegible] solution vector, $b_{\mathcal{D}} x_{\mathcal{D}}$, improves with respect to the fixed point $\phi_{\mathcal{D}}$ with a converging smoother as long as $x_{\mathcal{D}} > 0$ is not in the range of $(I - T_{\mathcal{D}}^{ML})^T$. For the solution to improve with respect to steady state vector $\tilde{\pi}_{\mathcal{D}}$ at each level, one requires sufficient conditions on the smoother, as in Theorem 4.16.

Then $x_{\mathcal{D}}$ at the finest level will improve from one ML iteration to the next, implying an improvement in the aggregated CTMC at each level and thus an improved solution at each level. Then, recalling from Lemma 4.21 that $\tilde{Q}_{\mathcal{D}} = Q_{\mathcal{D}}$ and $\rho(T_{\mathcal{D}}^{ML}) = 1$ upon convergence, $\rho(T_{\mathcal{D}}^{ML})$ and $\phi_{\mathcal{D}}$ must be approaching 1 and $\pi_{\mathcal{D}}$, respectively, while the subdominant eigenvalue of $T_{\mathcal{D}}^{ML}$ in magnitude is approaching zero with an increasing number of ML iterations.

In [11], extensive numerical experiments have been conducted with the ML solver on HMMs. Therein, the values chosen for the parameters of the $POWER$, $JOR$, and $SOR$ smoothers are $\alpha_{\mathcal{D}} = \max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|/0.999$ and $\omega = 1$, and the initial approximation is the uniform distribution. Furthermore, at least one pre- and one postsmoothing is performed at each level, and the coarsest system is solved using Gaussian elimination. Hence, $POWER$ is enforced to yield a converging smoother, and the $JOR$ and $SOR$ iteration matrices are nonnegative. Although $\omega = 1$ does not guarantee converging $JOR$ and $SOR$ smoothers (see Lemma 4.8), the results indicate that convergence may still be achieved. Hence, we conclude that the conditions stated in Theorem 4.16 for the smoothers are sufficient for convergence, but not necessary.

## 5. Experimental results.

In [13, sec. 5], we step through the ML method on Example 1 in section 2. Here we consider the number of iterations and the time in seconds required to reach $\|r\|_\infty < 10^{-8}$ (see Algorithm 1) for Examples 2 and 3. We compare SOR and ML methods with $(C, S, O) \in \{(V, SOR, FIXED), (W, SOR, CIRCULAR), (F, SOR, DYNAMIC)\}$. In all cases, the relaxation parameter of $SOR$ is set to 1. All experiments are performed on PCs with AMD opteron 2.3 GHz CPU and 1 GBytes of main memory.

Table 5.1

*Number of iterations and solution times for Example* 2.

| | SOR | | ML | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | $(V, SOR, FIXED)$ | | $(W, SOR, CIRCULAR)$ | | $(F, SOR, DYNAMIC)$ | |
| $K$ | *it* | *time* | *it* | *time* | *it* | *time* | *it* | *time* |
| 3 | 180 | 0 | 90 | 0 | 44 | 0 | 46 | 0 |
| 4 | 260 | 0 | 106 | 0 | 40 | 0 | 42 | 0 |
| 5 | 290 | 2 | 92 | 1 | 34 | 1 | 36 | 1 |
| 6 | 360 | 42 | 104 | 8 | 30 | 3 | 36 | 4 |
| 7 | 420 | 123 | 114 | 57 | 30 | 19 | 32 | 20 |

Table 5.1 contains the results for Example 2. For the solution we choose SOR and three variants of ML methods. For the latter we choose $\nu_1 = \nu_2 = 1$ in all cases. It can be seen that the number of iterations of SOR increases with an increasing number of LLMs. For the ML solver with $FIXED$ aggregation of LLMs, a small increase in the number of iterations can also be observed. For the other two ML solvers with $CIRCULAR$ and $DYNAMIC$ order of aggregating LLMs, the number of ML iterations does not increase, even becomes smaller with an increasing number of LLMs, and is much smaller than that of the corresponding $FIXED$ order. This behavior can be observed for all cycle types; it depends on the order of aggregation and shows the importance of modifying the order for this example. Hence, although convergence does not depend on the order of aggregating LLMs, rate of convergence does. It should be mentioned that this example is not particularly hard to solve with SOR since LLMs are strongly coupled and the number of iterations is fairly small. Nevertheless, the use of ML steps increases convergence speed significantly, reducing the solution times for the larger configurations by almost an order of magnitude.

The third example is much harder to solve with SOR or other classical iterative

TABLE 5.2
*Number of iterations and solution times for Example* 3.

| | SOR | | ML | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | $(V, SOR, FIXED)$ | | $(V, SOR, FIXED)$ | | $(W, SOR, CIRCULAR)$ | |
| | | | $(\nu_1 = 1,\ \nu_2 = 1)$ | | $(\nu_1 = 1,\ \nu_2 = 5)$ | | $(\nu_1 = 1,\ \nu_2 = 1)$ | |
| $K$ | *it* | *time* | *it* | *time* | *it* | *time* | *it* | *time* |
| 2 | 60 | 0 | 12 | 0 | 12 | 0 | 12 | 0 |
| 3 | 400 | 0 | 12 | 0 | 12 | 0 | 14 | 0 |
| 4 | 3,200 | 21 | 12 | 0 | 12 | 0 | 18 | 0 |
| 5 | 26,560 | 3,310 | 12 | 2 | 12 | 3 | 20 | 4 |
| 6 | | >10,000 | 18 | 45 | 12 | 42 | 14 | 38 |
| 7 | | | 16 | 492 | 12 | 554 | 14 | 529 |

methods. With the addition of a new LLM a new time scale is introduced in the model. It is known that such models are difficult to solve. Results for Example 3 are shown in Table 5.2. The results for SOR therein indicate that with an increasing number of LLMs the number of iterations grows drastically, and the system becomes practically unsolvable for $K > 5$. In the ML methods with fixed order of aggregation, at every aggregation step the fastest time scale is removed, and the system is mainly solved for the fastest remaining time scale. This implies that during a cycle each time scale is considered. Thus, we can expect fast convergence, which is confirmed by the results in Table 5.2. The number of iterations is almost independent of the number of LLMs, and even the largest configuration with $10,000,000$ states can be solved in less than 10 minutes, whereas SOR requires almost an hour to solve the system with only $10,000$ states. Since the ordering of LLMs is optimal according to the time scales, $CIRCULAR$ or $DYNAMIC$ ordering of aggregation do not help. The last two columns contain results for $CIRCULAR$ ordering and a $W$-cycle; results are similar to the $FIXED$ ordering. $DYNAMIC$ ordering gives worse results since the projected residuals which we use as a heuristic for choosing LLMs to be aggregated depend on the transition rates such that LLMs with small rates are aggregated first, resulting in a poor convergence in this example.

## 6. Conclusion.

In this paper, the convergence of a class of multilevel (ML) methods for large sparse Markov chains (MCs) has been investigated. The particular class of ML methods is inspired by algebraic multigrid and iterative aggregation-disaggregation, and has the capability of using (V, W, F) cycles, (power, Jacobi over relaxation (JOR), successive over relaxation (SOR)) methods as smoothers, and (fixed, circular, dynamic) orders in which coarser MCs can be formed by aggregation in a cycle. The conditions sufficient for convergence are an irreducible MC, a positive initial approximation from an appropriate subspace, an onto mapping of states from a finer MC to a coarser MC at each level, a uniformization parameter larger than the minimum magnitude of the diagonal elements for the power method, a relaxation parameter less than 1 for JOR and SOR, a sufficient number of pre- and postsmoothings at each level so as to ensure a smoothing matrix which is positive or has a(n almost) positive row/column, and the accurate solution of the coarsest system at each cycle. The asymptotic convergence rate of the class of ML methods across multiple levels is yet to be investigated.

**Acknowledgments.** We thank the anonymous referees for their constructive remarks and suggesting some of the references.

## REFERENCES

[1] F. Bause, P. Buchholz, and P. Kemper, *A toolbox for functional and quantitative analysis of DEDS*, in Quantitative Evaluation of Computing and Communication Systems, R. Puigjaner, N. N. Savino, and B. Serra, eds., Lecture Notes in Comput. Sci. 1469, Springer-Verlag, Heidelberg, Germany, 1998, pp. 356–359.

[2] R. Bellman, *Introduction to Matrix Analysis*, 2nd ed., Classics in Appl. Math. 19, SIAM, Philadelphia, PA, 1997.

[3] M. Benzi and T. Dayar, *The arithmetic mean method for finding the stationary vector of Markov chains*, Parallel Algorithms and Appl., 6 (1995), pp. 25–37.

[4] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Appl. Math. 9, SIAM, Philadelphia, PA, 1994.

[5] A. Borobia and J. Moro, *On nonnegative matrices similar to positive matrices*, Linear Algebra Appl., 266 (1997), pp. 365–379.

[6] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, PA, 2000.

[7] P. Buchholz, *Hierarchical structuring of superposed GSPNs*, IEEE Trans. Softw. Engrg., 25 (1999), pp. 166–181.

[8] P. Buchholz, *Structured analysis approaches for large Markov chains*, Appl. Numer. Math., 31 (1999), pp. 375–404.

[9] P. Buchholz, *Multilevel solutions for structured Markov chains*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 342–357.

[10] P. Buchholz and T. Dayar, *Block SOR for Kronecker structured Markovian representations*, Linear Algebra Appl., 386 (2004), pp. 83–109.

[11] P. Buchholz and T. Dayar, *Comparison of multilevel methods for Kronecker structured Markovian representations*, Computing, 73 (2004), pp. 349–371.

[12] P. Buchholz and T. Dayar, *Block SOR preconditioned projection methods for Kronecker structured Markovian representations*, SIAM J. Sci. Comput., 26 (2005), pp. 1289–1313.

[13] P. Buchholz and T. Dayar, *On the Convergence of a Class of Multilevel Methods for Large, Sparse Markov Chains*, Technical Report BU-CE-0601, Department of Computer Engineering, Bilkent University, Ankara, Turkey, 2006; available online from http://www.cs.bilkent.edu.tr/tech-reports/2006/BU-CE-0601.pdf.

[14] P. Buchholz, G. Ciardo, S. Donatelli, and P. Kemper, *Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models*, INFORMS J. Comput., 12 (2000), pp. 203–222.

[15] P. Buchholz and P. Kemper, *On generating a hierarchy for GSPN analysis*, Performance Eval. Rev., 26 (1998), pp. 5–14.

[16] P. Fernandes, B. Plateau, and W. J. Stewart, *Efficient descriptor-vector multiplications in stochastic automata networks*, J. ACM, 45 (1998), pp. 381–414.

[17] P. Fernandes, B. Plateau, and W. J. Stewart, *Optimizing tensor product computations in stochastic automata networks*, RAIRO Oper. Res., 32 (1998), pp. 325–351.

[18] A. Greenbaum, *Analysis of a multigrid method as an iterative technique for solving linear systems*, SIAM J. Numer. Anal., 21 (1984), pp. 473–485.

[19] G. Horton and S. Leutenegger, *A multi-level solution algorithm for steady state Markov chains*, Performance Eval. Rev., 22 (1994), pp. 191–200.

[20] U. Krieger, *Numerical solution of large finite Markov chains by algebraic multigrid techniques*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer Academic Publishers, Boston, MA, 1995, pp. 403–424.

[21] I. Marek and P. Mayer, *Convergence analysis of an iterative aggregation/disaggregation method for computing stationary probability vectors of stochastic matrices*, Numer. Linear Algebra Appl., 5 (1998), pp. 253–274.

[22] I. Marek and I. Pultarova, *A note on local and global convergence analysis of iterative aggregation-disaggregation methods*, Linear Algebra Appl., 413 (2006), pp. 327–341.

[23] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, PA, 2000.

[24] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., Frontiers in Appl. Math. 3, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[25] U. Rüde, *The Multigrid Workbench*, http://www.mgnet.org/mgnet/tutorials/xwb.html.

[26] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.

[27] S. Serra, *Multi-iterative methods*, Comput. Math. Appl., 26 (1993), pp. 65–87.

[28] G. W. Stewart, *Matrix Algorithms, Vol.* I: *Basic Decompositions*, SIAM, Philadelphia, PA, 1998.

[29] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

[30] J.-P. Tremblay and R. Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw–Hill, New York, 1975.
[31] E. Uysal and T. Dayar, *Iterative methods based on splittings for stochastic automata networks*, European J. Oper. Res., 110 (1998), pp. 166–186.
[32] C. F. Van Loan, *The ubiquitous Kronecker product*, J. Comput. Appl. Math., 123 (2000), pp. 85–100.
[33] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, New York, 1992.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 4, pp. 1051–1064

# APPROXIMATE DIAGONALIZATION[*]

E. B. DAVIES[†]

**Abstract.** We describe a new method of computing functions of highly nonnormal matrices by using the concept of approximate diagonalization. We formulate a conjecture about its efficiency and provide both theoretical and numerical evidence in support of the conjecture. We apply the method to compute arbitrary real powers of highly nonnormal matrices.

**Key words.** Jordan matrices, ill-posedness, regularization, fractional powers, functional calculus, spectral theory

**AMS subject classifications.** 65F22, 65F15, 15A60, 15A42, 47A52, 47A60

**DOI.** 10.1137/060659909

## 1. Introduction.

Let $A$ be a nonnormal $n \times n$ matrix and suppose that one wants to evaluate $x = f(A)b$ or solve $f(A)x = b$ for a large number of different analytic functions $f$ rapidly, without caring too much about high accuracy. If $A$ is diagonalizable, i.e., $A := SDS^{-1}$, where $D$ is diagonal, then one can solve the first problem by writing $x := Sf(D)S^{-1}b$, where $f(D)$ is evaluated by applying the function $f$ to the diagonal entries of $D$, which coincide with the eigenvalues of $A$. The second problem may be solved in a similar manner.

This procedure may not be appropriate if $A$ is highly nonnormal, because the eigenvalues of $A$ can be highly unstable under small perturbations, such as those associated with rounding errors in computation, and the matrix $S$ may have an extremely large condition number $\kappa(S) := \|S\|\,\|S^{-1}\|$. In the most extreme case, when $A$ has a nontrivial Jordan form, the method breaks down entirely.

In this paper we describe an approach which involves using an approximate diagonalization of $A$. We emphasize that this does not mean that it is close to a true diagonalization, but rather that it has many of the features of a true diagonalization, and the amount of error associated with using it can be estimated. We start by describing the idea and formulate a conjecture about its efficiency. Much of the remainder of this paper is devoted to providing theoretical and numerical evidence in support of the conjecture.

In section 5 we use the ideas developed to throw some light on the difficulties of computing fractional powers of matrices that are close to singular.

## 2. Definitions.

Throughout this paper we assume that $f$ is an analytic function defined on a neighborhood of $\mathrm{Spec}(A)$ and that $f(A)$ is defined by means of the holomorphic functional calculus as in [1, section 1.5]. In order to be able to define $f(A)$ stably for highly nonnormal $A$ one has to impose some conditions on the analytic function $f$. If $f(z) = (z-a)^{-1}$, where $a$ is not close to $\mathrm{Spec}(A)$, one may nevertheless have $a \in \mathrm{Spec}(\tilde{A})$, where $\|A - \tilde{A}\|$ is very small. In such situations $f(A)$ cannot be defined stably for reasons discussed in [11].

---

[*] Received by the editors May 15, 2006; accepted for publication (in revised form) by N. J. Higham March 20, 2007; published electronically November 2, 2007. This work was supported under EPSRC grant GR/R81756.

http://www.siam.org/journals/simax/29-4/65990.html

[†] Department of Mathematics, King's College of London, Strand, London, WC2R 2LS, UK (E.Brian.Davies@kcl.ac.uk).

We assume henceforth that $\gamma$ is a simple closed curve with length $|\gamma|$ and that $\mathrm{Spec}(A) \subseteq U$, where $U$ is the region inside $\gamma$. We also assume that $f \in \mathcal{A}$, where $\mathcal{A}$ is the space of functions that are analytic on $\gamma \cup U$. We write $\|f\|_\infty$ for the maximum value of $|f(z)|$ for $z \in \gamma$, or equivalently for $z \in (\gamma \cup U)$.

LEMMA 2.1. [illegible] $R(z, A) := (zI - A)^{-1}$ [illegible] $\|R(z, A)\| \le c$ [illegible] $z \in \gamma$ [illegible] $f \in \mathcal{A}$ [illegible]

$$\|f(A)\| \le \frac{c}{2\pi} |\gamma| \, \|f\|_\infty.$$

[illegible] $\|A - \tilde{A}\| \le 1/(2c)$ [illegible] $\tilde{f} \in \mathcal{A}$ [illegible]

$$\|f(A) - \tilde{f}(\tilde{A})\| \le c^2 \pi^{-1} |\gamma| \, \|f\|_\infty \, \|A - \tilde{A}\| + c\pi^{-1} |\gamma| \, \|f - \tilde{f}\|_\infty.$$

[illegible]. The first bound depends on a routine estimate of the formula

$$f(A) = \frac{1}{2\pi i} \int_\gamma f(z) R(z, A) \, \mathrm{d}z.$$

We next assume that $z \in \gamma$ and use the bound

$$\begin{aligned} \|R(z, \tilde{A})\| &= \left\| R(z, A) \left( I - (\tilde{A} - A) R(z, A) \right)^{-1} \right\| \\ &\le \frac{\|R(z, A)\|}{1 - \|(A - \tilde{A}) R(z, A)\|} \\ &\le 2c \end{aligned}$$

to derive

$$\begin{aligned} \|R(z, \tilde{A}) - R(z, A)\| &= \|R(z, \tilde{A})(\tilde{A} - A) R(z, A)\| \\ &\le 2c^2 \|A - \tilde{A}\|. \end{aligned}$$

The second formula now follows by a routine estimate of the identity

$$\begin{aligned} f(A) - \tilde{f}(\tilde{A}) &= \frac{1}{2\pi i} \int_\gamma f(z) \left( R(z, A) - R(z, \tilde{A}) \right) \mathrm{d}z \\ &\quad + \frac{1}{2\pi i} \int_\gamma (f(z) - \tilde{f}(z)) R(z, \tilde{A}) \, \mathrm{d}z. \qquad \square \end{aligned}$$

[illegible] 2.2. There are two obvious ways of ensuring that the resolvent bound of Lemma 2.1 is satisfied. The first uses the stability of the numerical range $\mathrm{Num}(A)$ under small perturbations. If $\mathrm{Num}(A) \subseteq U$ and $\mathrm{dist}(\gamma, \mathrm{Num}(A)) \ge 1/c$, then the bound $\|R(z, A)\| \le c$ is valid for all $z \in \gamma$ by [11, Chapter 17] or [1, section 9.3].

Alternatively, given $c > 0$ one may define $\gamma$ to be the pseudospectral contour $\{z : \|R(z, A)\| = c\}$. The shape of the contour, which may have several components, can be determined numerically by using the Eigtool software; see [12].

We say that three matrices $S, D, B$ provide an approximate diagonalization of $A$ if $D$ is diagonal, $S$ is invertible, $B$ is small, and $A = SDS^{-1} + B$; we assume that $\|A\| \le 1$ whenever necessary for reasons stated below. We say that $S, B$ is a permitted pair for $A$ if $S$ is invertible and $D := S^{-1}(A - B)S$ is diagonal. The accuracy of the approximate diagonalization is measured by the quantity

$$\sigma(A, S, B, \varepsilon) := \kappa(S)\varepsilon + \|B\|,$$

where $\varepsilon \in (0, 1)$ is a preassigned degree of accuracy of the computations, for example, $\varepsilon := 10^{-16}$. Note that $1 = \|SS^{-1}\| \leq \|S\|\,\|S^{-1}\| = \kappa(S)$ for all $S$. The term $\kappa(S)\varepsilon$ measures errors associated with the condition number of $S$ and would vanish if the computations could have infinite precision, i.e., if $\varepsilon = 0$. The term $\|B\|$ represents the amount that $A$ has been perturbed with the intention of reducing the first type of error. By adding the two errors and then minimizing over all permitted pairs, one obtains the smallest overall error that is possible when diagonalizing $A$ approximately, namely

$$\underline{\sigma}(A, \varepsilon) := \inf_{B,S} \sigma(A, S, B, \varepsilon).$$

The nonzero entries of $D$ are the eigenvalues of $A - B$ and are of order 1 in all the cases considered below.

After choosing the perturbation $B$ the matrix $D$ is uniquely determined up to a permutation of its diagonal entries. However, one may replace $S$ by $SE$ where $E$ is any invertible matrix that commutes with $D$, for example, any invertible diagonal matrix. Different choices of $S$ may have very different condition numbers. Fortunately, the MATLAB algorithm `[S,D]=eig(A)` is known to choose a matrix $S$ whose condition number is within a factor $\sqrt{n}$ of the best (i.e., smallest) value possible [9, section 7.3], [10]; for the derogatory case, however, see [4].

Many of our theorems below can be viewed as providing support for the following conjecture.

[illegible] For every positive integer $n$ there exists $c_n$ such that

$$\underline{\sigma}(A, \varepsilon) \leq c_n \varepsilon^{1/2}$$

for every $n \times n$ matrix $A$ such that $\|A\| \leq 1$ and for every $\varepsilon \in (0, 1)$.

Since one can only evaluate $\underline{\sigma}(A, \varepsilon)$ exactly in simple cases, we attempt to obtain a fairly sharp upper bound on it by choosing $B, S$ appropriately. The rate of convergence of $\underline{\sigma}(A, \varepsilon)$ to 0 as $\varepsilon \to 0$ depends on whether $A$ is diagonalizable or not. Note that one obtains an approximate diagonalization for another matrix $\tilde{A}$ from that for $A$ by keeping the same $S, D$ and putting $\tilde{B} := B + (\tilde{A} - A)$. Therefore

$$|\underline{\sigma}(A, \varepsilon) - \underline{\sigma}(\tilde{A}, \varepsilon)| \leq \|A - \tilde{A}\|$$

and our definition is computationally stable. Further computational questions can be asked, for example, about the errors arising when evaluating $S^{-1}$ for a choice of $S$ that is close to singular, but the methods described here allow one to replace $S^{-1}$ by $T$ provided

$$\frac{\|T - S^{-1}\|}{\|S^{-1}\|} = O(\varepsilon).$$

We observe that

$$\underline{\sigma}(VAV^{-1}, \varepsilon) \leq \kappa(V)\underline{\sigma}(A, \varepsilon) \tag{2.1}$$

for all invertible matrices $V$; thus the order of magnitude of $\underline{\sigma}(A,\varepsilon)$ is not changed if one passes from $A$ to $VAV^{-1}$, where $\kappa(V)$ is of order 1. If $A$ is normal, then one may diagonalize it exactly with $S$ unitary and $B=0$, so $\underline{\sigma}(A,\varepsilon)=\varepsilon$.

A feature of our definitions of $\sigma$ and $\underline{\sigma}$ is that they do not scale under the map $A\to\lambda A$ when $\lambda$ is large. As $\lambda$ increases, $\mathrm{Spec}(\lambda A)$ and $\mathrm{Num}(\lambda A)$ expand, so the contour $\gamma$ and the algebra $\mathcal{A}$ must be changed. We therefore impose the condition $\|A\|\le 1$ whenever necessary.

The function $\underline{\sigma}(A,\varepsilon)$ is closely related to $\mu(A,\delta)$ defined for all $\delta>0$ by

$$\mu(A,\delta):=\inf\{\kappa(S): A=SDS^{-1}+B,\text{ where } D \text{ is diagonal and } \|B\|\le\delta\}.$$

A simple compactness argument implies that the infimum is actually attained.

LEMMA 2.3. [illegible] $c>0$ [illegible] $\alpha>0$ [illegible] $\mu(A,\delta)\le c\delta^{-\alpha}$ [illegible] $\delta\in(0,1)$ [illegible]

$$\underline{\sigma}(A,\varepsilon)\le 2(c\varepsilon)^{1/(\alpha+1)}$$

[illegible] $\varepsilon\in(0,1/c)$

[illegible]. If the infimum in the definition of $\mu(A,\delta)$ is attained for $A,S,D,B$, then

$$\underline{\sigma}(A,\varepsilon)\le\sigma(A,S,B,\varepsilon)\le\mu(A,\delta)\varepsilon+\delta\le c\varepsilon\delta^{-\alpha}+\delta$$

for all $\delta>0$. The lemma follows by applying the following general fact: If $f$ (resp., $g$) is a nonnegative, monotonically decreasing (resp., increasing) function on $(a,b)$ and $f(\xi)=g(\xi)$ for some $\xi\in(a,b)$, then

$$f(\xi)\le\inf\{f(x)+g(x):x\in(a,b)\}\le 2f(\xi).$$

THEOREM 2.4. [illegible] $\|A\|\le 1$ [illegible] $f(z)$ [illegible] $\{z:|z|\le r\}$ [illegible] $r>1$ [illegible] $\underline{\sigma}(A,\varepsilon)<(r-1)/2$ [illegible]

$$\underline{\sigma}(f(A),\varepsilon)<\underline{\sigma}(A,\varepsilon)\max\left\{1,\frac{2r\|f\|_{r,\infty}}{(r-1)^2}\right\},$$

[illegible]

$$\|f\|_{r,\infty}:=\max\{|f(z)|:|z|\le r\}.$$

[illegible]. If $\tilde{A}:=SDS^{-1}$, then $\|A-\tilde{A}\|=\|B\|\le\sigma(A,S,B,\varepsilon)$ and we can define $\tilde{B}$ by

$$f(A)=f(\tilde{A})+\tilde{B}=Sf(D)S^{-1}+\tilde{B}.$$

If $r>1$ and $\gamma$ is the circle $\{z:|z|=r\}$, then $\|R(z,A)\|\le(r-1)^{-1}$ for all $z\in\gamma$. Lemma 2.1 implies that if $\sigma(A,S,B,\varepsilon)<(r-1)/2$, then

$$\|\tilde{B}\|\le\|B\|\,\frac{2r\|f\|_{r,\infty}}{(r-1)^2}.$$

Therefore

$$\sigma(f(A),S,\tilde{B},\varepsilon)=\kappa(S)\varepsilon+\|\tilde{B}\|\le\sigma(A,S,B,\varepsilon)\max\left\{1,\frac{2r\|f\|_{r,\infty}}{(r-1)^2}\right\}.$$

The theorem now follows by taking the infimum over all permitted $S,B$.

We next establish a close connection between the above ideas and the existence of a suitable basis of (column) pseudo-eigenvectors. The results obtained are only of interest when $\|B\|$ and all $\|r_j\|$ are very small.

THEOREM 2.5. [illegible] $A = SDS^{-1} + B$ [illegible] $D$ [illegible] $\lambda_1, \dots, \lambda_n$ [illegible] $j$ [illegible] $\phi_j$ [illegible] $S$ [illegible] $j$ [illegible]

$$A\phi_j = \lambda_j \phi_j + r_j, \tag{2.2}$$

[illegible]

$$\|r_j\| \le \|B\|$$

[illegible] $j$

[illegible]. Applying the identity $AS = SD + BS$ to the standard basis elements $\{e_1, \dots, e_n\}$ of $\mathbf{C}^n$ yields (2.2) with $r_j := B\phi_j$. The bound follows immediately.

The above theorem has the following partial converse.

THEOREM 2.6. [illegible] $\{\phi_1, \dots, \phi_n\}$ [illegible] $\mathbf{C}^n$ [illegible] $\|\phi_j\| = 1$ [illegible]

$$A\phi_j = \lambda_j \phi_j + r_j \tag{2.3}$$

[illegible] $j \in \{1, \dots, n\}$ [illegible] $\lambda_j \in \mathbf{C}$ [illegible] $r_j \in \mathbf{C}^n$ [illegible] $R, S$ [illegible] $n \times n$ [illegible] $R := [r_1 \dots r_n]$ [illegible] $S := [\phi_1 \dots \phi_n]$ [illegible] $A = SDS^{-1} + B$ [illegible] $B := RS^{-1}$ [illegible] $D$ [illegible] $\lambda_1, \dots, \lambda_n$ [illegible]

$$1 \le \|S\| \le \sqrt{n}$$

[illegible]

$$\|B\| \le \|S^{-1}\| \left\{ \sum_{j=1}^{n} \|r_j\|^2 \right\}^{1/2}.$$

[illegible]. Equation (2.3) may be rewritten in the form $AS = SD + R$ by concatenating the columns. This implies $B = RS^{-1}$. We also have

$$1 \le \|S\| \le \|S\|_{\mathrm{HS}} = \sqrt{n},$$

where $\|\cdot\|_{\mathrm{HS}}$ is the Hilbert–Schmidt, or Frobenius, norm. Similarly,

$$\|B\| \le \|S^{-1}\|\,\|R\| \le \|S^{-1}\|\,\|R\|_{\mathrm{HS}} = \|S^{-1}\| \left\{ \sum_{j=1}^{n} \|r_j\|^2 \right\}^{1/2}.$$

## 3. Evidence supporting the conjecture.

We start by proving our conjecture for Jordan matrices.

LEMMA 3.1. [illegible] $J$ [illegible] $n \times n$ [illegible]

$$J_{r,s} := \begin{cases} 1 & \text{[illegible] } s = r + 1, \\ 0 & \text{[illegible]}. \end{cases} \tag{3.1}$$

[illegible]

$$1 \le \mu(J, \delta) \le \delta^{-1+1/n} \le \delta^{-1} \tag{3.2}$$

[illegible] $\delta \in (0,1)$ [illegible]

$$0 \le \underline{\sigma}(J,\varepsilon) \le 2\varepsilon^{n/(2n-1)} \le 2\varepsilon^{1/2} \tag{3.3}$$

[illegible] $\varepsilon \in (0,1)$

[illegible] We define $B$ by

$$B_{r,s} := \begin{cases} -\delta & \text{if } r = n \text{ and } s = 1, \\ 0 & \text{otherwise,} \end{cases}$$

and $T$ by

$$T_{r,s} := \begin{cases} \delta^{-r/n} & \text{if } r = s, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

A direct calculation shows that

$$T(J-B)T^{-1} = \delta^{1/n}U,$$

where $U$ is the circulant and unitary matrix with entries

$$U_{r,s} := \begin{cases} 1 & \text{if } s = r+1, \\ 1 & \text{if } r = n \text{ and } s = 1, \\ 0 & \text{otherwise.} \end{cases}$$

If $F$ is the finite Fourier transform whose matrix

$$F_{r,s} := n^{-1/2}\mathrm{e}^{2\pi i rs/n}$$

is unitary, then

$$FUF^{-1} = D,$$

where $D$ is the diagonal matrix with entries $\lambda_r = \mathrm{e}^{2\pi i r/n}$, for $1 \le r \le n$. Putting $S := FT$ we finally obtain

$$J = S^{-1}(\delta^{1/n}D)S + B.$$

Since $\|T\| = \delta^{-1}$ and $\|T^{-1}\| = \delta^{1/n}$ we deduce that $\kappa(S) = \delta^{-1+1/n}$. This implies (3.2). The corresponding upper bound on $\underline{\sigma}$ is obtained by applying Lemma 2.3.

[illegible] 3.2. We compare the above theoretical result with what can be obtained numerically. We defined $J$ by (3.1) with $n = 25$ and evaluated $f(\delta) := \delta^{1-1/n}\kappa(S)$ for 200 randomly generated matrices $B$ with norms equal to $\delta$ for a range of values of $\delta$. The matrices $S$ and $D$ were defined by using the MATLAB command `[S,D]=eig(A-B)`. In Table 3.1, $\min(f(\delta))$ is the minimum value of $f(\delta)$ obtained and $\mathrm{med}(f(\delta))$ is the median value. We also took a sample of 2000 such matrices $B$ and found that all the values of $\min(f(\delta))$ remained larger than 2. The similarity of the numerical results to what was proved in Lemma 3.1 suggests that both are close to the optimal bound.

The following corollary does not prove the conjecture because the constant obtained depends on the matrix involved, not just on the dimension. It is known that finding the Jordan canonical form is an inherently unstable problem [6, p. 390], [7].

Table 3.1
*Computation of condition numbers in Example* 3.2.

| $\delta$ | $\min(f(\delta))$ | $\mathrm{med}(f(\delta))$ |
|---|---|---|
| $10^{-1}$ | 3.67 | 17.91 |
| $10^{-2}$ | 4.15 | 22.09 |
| $10^{-3}$ | 3.29 | 22.25 |
| $10^{-4}$ | 3.78 | 25.57 |
| $10^{-5}$ | 3.88 | 25.95 |
| $10^{-6}$ | 3.32 | 24.04 |
| $10^{-7}$ | 3.83 | 20.51 |
| $10^{-8}$ | 3.72 | 24.99 |

Corollary 3.3. [illegible] $n \times n$ [illegible] $A$ [illegible] $c_A$ [illegible]

$$\underline{\sigma}(A,\varepsilon) \le c_A \varepsilon^{1/2}$$

[illegible] $\varepsilon \in (0,1)$

[illegible]. If $A = V\tilde{J}V^{-1}$, where $\tilde{J}$ is a Jordan canonical form for $A$, then (2.1) implies that

$$\underline{\sigma}(A,\varepsilon) \le \kappa(V)\underline{\sigma}(\tilde{J},\varepsilon).$$

By applying the method of Lemma 3.1 to each Jordan block of $\tilde{J}$, we obtain the corollary.

We next prove the conjecture for triangular Toeplitz matrices.

Theorem 3.4. [illegible] $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ [illegible] $\sum_{r=0}^{n-1} |\alpha_r| \le 1$ [illegible] $A$ [illegible] $n \times n$ [illegible]

$$A_{r,s} := \begin{cases} \alpha_{s-r} & \text{[illegible] } s \ge r, \\ 0 & \text{[illegible]}. \end{cases}$$

[illegible]

$$\underline{\sigma}(A,\varepsilon) \le 2\varepsilon^{n/(2n-1)} \le 2\varepsilon^{1/2}$$

[illegible] $\varepsilon \in (0,1)$

[illegible]. If we define $B$ by

$$B_{r,s} := \begin{cases} -\delta\alpha_{s-r+n} & \text{if } s < r, \\ 0 & \text{otherwise,} \end{cases}$$

where $\delta \in (0,1)$ is to be determined, then $\|B\| \le \delta$. If we define $T$ by (3.4), then a direct calculation shows that

$$C := T(A-B)T^{-1}$$

is the circulant matrix with entries

$$C_{r,s} := \alpha_{s-r}\delta^{(s-r)/n},$$

where we replace $s-r$ by $s-r+n$ if the former expression is negative. If $F$ is the finite Fourier transform, then

$$D := FCF^{-1}$$

is a diagonal matrix. Putting $S := FT$ as before we obtain $\kappa(S) = \delta^{-1+1/n}$ and

$$A = S^{-1}DS + B.$$

Putting $\delta := \varepsilon^{n/(2n-1)}$ we obtain

$$\underline{\sigma}(A, \varepsilon) < 2\varepsilon^{n/(2n-1)}.$$

We have not been able to prove the conjecture for general $n \times n$ matrices; Theorem 3.9 below is the closest that we have got to it. We originally proved it under the assumption that the eigenvalues of $A$ were collinear. The general case depends on the following theorem of Friedland [5]. Its proof depends on using the degree mod 2 of a smooth map between manifolds of equal dimension, and it would be valuable to obtain a constructive version. This may not be easy, because the number of normal "extensions" $N$ of $Q$ varies from 1 to $\infty$ (inclusive) depending on $Q$.

THEOREM 3.5 (see Friedland [5]). [illegible] $n \times n$ [illegible] $Q$ [illegible] $L$ [illegible] $N := Q + L$ [illegible]

[illegible] 3.6. A direct construction of the matrix $L$ in the theorem is not elementary even in the case of $3 \times 3$ matrices. If the eigenvalues of the upper triangular matrix $Q$ are collinear, then we may construct $L$ as follows. We first write $Q$ in the form $Q = cI + \mathrm{e}^{i\theta}(D + U)$, where $c \in \mathbf{C}$, $\theta \in \mathbf{R}$, $D$ is a real diagonal matrix, and $U$ is strictly upper triangular. If we define $L := \mathrm{e}^{i\theta}U^*$, then $Q + L = cI + \mathrm{e}^{i\theta}H$, where $H$ is self-adjoint, and this implies that $Q + L$ is normal.

LEMMA 3.7. [illegible] $N := Q + L$ [illegible] $Q$ [illegible] $L$ [illegible] $\nu(Q) = \nu(L)$ [illegible]

$$\nu(A) := \sum_{r,s} |r - s||A_{r,s}|^2.$$

[illegible] We have

$$\begin{aligned} \nu(Q) - \nu(L) &= \sum_{r,s} (s - r)|N_{r,s}|^2 \\ &= \mathrm{tr}[N^*EN] - \mathrm{tr}[N^*NE] \\ &= \mathrm{tr}[(NN^* - N^*N)E] \\ &= 0, \end{aligned}$$

where

$$E_{r,s} := \begin{cases} r & \text{if } r = s, \\ 0 & \text{otherwise.} \end{cases}$$

Further comparisons between the size of $Q$ and $L$ can be obtained by replacing $E$ by

$$E_{r,s} := \begin{cases} f(r) & \text{if } r = s, \\ 0 & \text{otherwise,} \end{cases}$$

where $f$ is any monotonic function on $\{1, \ldots, n\}$.

LEMMA 3.8. [illegible] $\nu(A) \le n^2\|A\|^2$ [illegible] $n \times n$ [illegible] $A$ [illegible] $A$ [illegible] $\|A\|^2 \le \nu(A)$

[illegible]. We have

$$\nu(A) \le n \sum_{r,s} |A_{r,s}|^2 = n \sum_{s=1}^{n} \|Ae_s\|^2 \le n^2 \|A\|^2,$$

where $\{e_s\}_{s=1}^n$ is the standard basis of $\mathbf{C}^n$.

If the diagonal entries of $A$ vanish, then the second inequality follows from

$$\|A\|^2 \le \|A\|_{\mathrm{HS}}^2 \le \nu(A).$$

Theorem 3.9. [illegible] $A$ [illegible] $n \times n$ [illegible] $\|A\| \le 1$ [illegible]

$$\underline{\sigma}(A, \varepsilon) \le (1+n)\varepsilon^{2/(n+1)}$$

[illegible] $\varepsilon \in (0,1)$ [illegible]

$$\underline{\sigma}(A, \varepsilon) < 4\varepsilon^{1/2}$$

[illegible] $3 \times 3$ [illegible] $A$ [illegible] $\|A\| \le 1$

[illegible]. By the Schur decomposition there exists a unitary matrix $U$ such that $P := U^{-1}AU$ is upper triangular. If $0 < \delta < 1$ and

$$V_{r,s} := \begin{cases} \delta^r & \text{if } r = s, \\ 0 & \text{otherwise,} \end{cases}$$

then $Q := V^{-1}PV$ is upper triangular and

$$\nu(Q) \le \delta^2 \nu(P) \le \delta^2 n^2 \|P\|^2 \le \delta^2 n^2.$$

By Friedland's theorem there exists a strictly lower triangular matrix $L$ such that $Q + L$ is normal and $\nu(L) = \nu(Q)$. A direct calculation establishes that

$$\|VLV^{-1}\|^2 \le \nu(VLV^{-1}) \le \delta^2 \nu(L) = \delta^2 \nu(Q) \le \delta^4 n^2.$$

Therefore $B := -UVLV^{-1}U^{-1}$ satisfies $\|B\| \le \delta^2 n$. Hence

$$V^{-1}U^{-1}(A - B)UV = Q + L = WDW^{-1},$$

where $D$ is diagonal and $W$ is unitary. Putting $S := UVW$ we obtain $A = SDS^{-1} + B$, where $\kappa(S) = \kappa(V) = \delta^{1-n}$. Therefore

$$\underline{\sigma}(A, \varepsilon) \le \delta^{1-n}\varepsilon + n\delta^2.$$

The result now follows by putting $\delta := \varepsilon^{1/(n+1)}$.

**4. Random perturbations.** The above methods of constructing $B$ and $S$ are too simple to prove the conjecture for $n > 3$. In this section we describe a randomized approximate diagonalization method (RADM), suggested to us by L. N. Trefethen, which provides numerical evidence in support of the conjecture. Numerically it is remarkably effective.

If the $n \times n$ matrix $A$ cannot be diagonalized or can only be diagonalized by means of a matrix $S$ whose condition number is extremely large, then one can instead diagonalize the matrix $A - B$, where $B$ is a small random perturbation. We found experimentally that for a variety of strictly upper triangular $n \times n$ matrices $A$ (none of which can be diagonalized) with $n = 100$ and $\varepsilon := 10^{-16}$ one has $\underline{\sigma}(A, \varepsilon) \le 3 \times 10^{-7}$. In each case we minimized over 100 randomly chosen $B$ such that $\|B\| = 10^{-8}$. On the other hand, for a series of 100 matrices of the form `A=rand(n)` with $n = 100$ and $B = 0$ we found that $50 \le \kappa(S) \le 1000$ in every case; our methods are not necessary for such matrices. In the computations that follow the random perturbation was of the form `B=s*randn(n)`, where $s$ is a small constant. However, we got the same results with small random perturbations `B=s*randn(n,1)*randn(1,n)` of rank one.

One can use RADM to evaluate $A^\alpha$ and other similar functions of $A$. Our conclusion from a range of such problems, some described below, is that RADM is less accurate than standard MATLAB algorithms when the matrix $A$ is quite close to being normal. If $A$ is far from normal and has a small eigenvalue, then the two methods have comparable accuracy. For many functions one cannot apply the MATLAB algorithm `funm`, described in [2], but RADM still yields a result whose accuracy can be confirmed by repeating the computation with another choice of the random perturbation.

[illegible] 4.1. We consider the $n \times n$ matrix

$$A_{r,s} := \begin{cases} r/n & \text{if } s = r+1, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

We put $n := 100$ and defined

```
B1=randn(n)
B=10^(-u)*B1/norm(B1)
[S,D] = eig(A-B)
```

in the notation of MATLAB. We computed $\sigma(A, S, B, \varepsilon)$ and $\log_{10}(\kappa(S))$ for $\varepsilon = 10^{-16}$ and $1 \le u \le 15$. Table 4.1 shows that $\sigma(A, S, B, \varepsilon)$ took its minimum value for $\|B\| \sim 10^{-7}$ but that the condition number of $S$ increased steadily as $u$ increased. The minimum value of $\sigma$ is of order $\varepsilon^{1/2}$.

We also carried out a computation in which the entries $r/n$ in (4.1) were replaced by randomly chosen numbers. The conclusions were similar.

TABLE 4.1
*Computation of condition numbers in Example* 4.1.

| $u$ | $\sigma(A, S, B, \varepsilon)$ | $\log_{10}(\kappa(S))$ |
|---|---:|---|
| 1 | 0.1 | 2.2784 |
| 2 | 0.01 | 3.723 |
| 3 | 0.001 | 4.3007 |
| 4 | 0.0001 | 5.3355 |
| 5 | $1e - 005$ | 6.169 |
| 6 | $1.0016e - 006$ | 7.1996 |
| 7 | $1.2316e - 007$ | 8.3647 |
| 8 | $2.0592e - 007$ | 9.2921 |
| 9 | $1.7551e - 006$ | 10.244 |
| 10 | $2.0103e - 005$ | 11.303 |
| 11 | 0.00015367 | 12.187 |
| 12 | 0.0015981 | 13.204 |
| 13 | 0.019479 | 14.29 |
| 14 | 0.19699 | 15.294 |
| 15 | 1.7837 | 16.251 |

**5. Fractional powers.** The definition of the square root of an $n \times n$ matrix $A$ is not as straightforward as it appears. If $A$ has $n$ distinct nonzero eigenvalues, then it has exactly $2^n$ square roots, which commute pairwise. On the other hand, the matrices 0 and 1 have a continuum of noncommuting square roots. If $A^n = 0$ but $A^{n-1} \neq 0$, then $A$ has no square root, but $A^2$ has a continuum of commuting square roots, namely $A + cA^{n-1}$ for any choice of $c$. If $A$ has $n$ distinct nonzero eigenvalues but two (or more) of these are approximately equal, then it may have a large number of pairwise noncommuting approximate square roots. One may avoid these ambiguities by using the holomorphic functional calculus to define $A^{1/2}$ and choosing the branch of $z^{1/2}$ that has a cut along the negative real axis.

[illegible] 5.1. Let $A$ be the $n \times n$ matrix

$$(A)_{r,s} := \begin{cases} r/n & \text{if } s = r+1, \\ c & \text{if } s = r, \\ 0 & \text{otherwise,} \end{cases} \tag{5.1}$$

where $0 < c < 1$. All such matrices satisfy $\|A\| \leq 2$ rather than $\|A\| \leq 1$ so a slight adjustment of the theory is needed. We computed $A^{1/2}$ for various values of $c$ when $n := 20$ by two methods and presented the results in Table 5.1. We compared $\|E^2 - A\|$, where $E$ is the square root computed by using RADM, with $\|F^2 - A\|$, where `F=sqrtm(A)` in the notation of MATLAB.

If one only looks at the third column of Table 5.1 one sees that the algorithm `sqrtm` is not accurate for $c < 0.1$. The situation for RADM is not as straightforward. Column 2 wrongly suggests that RADM is reasonably accurate for all values of $c$ investigated. However, column 4 shows that $\|E\|$ increases rapidly as $c$ decreases, while column 5 presents the ratio of two values of $\|E\|$ computed with two different random perturbations $B_i$, both satisfying $\|B_i\| = 10^{-8}$. One sees that the two values of $\|E\|$ obtained are quite different if $c < 0.2$, even though $\|E^2 - A\| = O(10^{-8})$ in both cases. This is possible because the map $A \to A^{1/2}$ varies very rapidly for small values of $c$ in spite of Lemma 2.1. If one only wants an approximate square root of $A$, then RADM works well for all $c$ investigated, but it does not produce a good approximation to the exact square root of $A$ for $c < 0.2$.

Our intention above was to illuminate the problems involved in computing square roots rather than to advocate the use of a particular method, but if one wishes to use RADM it is recommended that one should check that two successive applications with different random perturbations yield the same answer to within $O(10^{-8})$.

TABLE 5.1
*Computation of square roots in Example* 5.1.

| | RADM | sqrtm | | |
|---|---|---|---|---|
| $c$ | $\|E^2 - A\|$ | $\|F^2 - A\|$ | $\|E\|$ | $\|E_1\|/\|E_2\|$ |
| 0.8 | $3.0453e - 008$ | $1.8228e - 016$ | 1.2743 | 1 |
| 0.6 | $5.5242e - 008$ | $1.9563e - 016$ | 1.1984 | 1 |
| 0.4 | $1.5797e - 008$ | $5.6362e - 016$ | 5.3628 | 1 |
| 0.3 | $6.2193e - 008$ | $1.3148e - 014$ | 103.7 | 1.0001 |
| 0.2 | $2.6138e - 008$ | $3.2596e - 012$ | $2.0706e + 004$ | 0.97515 |
| 0.1 | $2.0012e - 008$ | $6.3565e - 008$ | $3.3292e + 006$ | 0.50668 |
| 0.05 | $2.8685e - 008$ | 0.0074873 | $3.0604e + 006$ | 0.55402 |
| 0.02 | $2.2364e - 008$ | 93768 | $9.1859e + 006$ | 0.71102 |
| 0.01 | $1.453e - 008$ | $5.5561e + 008$ | $2.2661e + 007$ | 0.89547 |

Let $A$ be an $n \times n$ matrix whose numerical range is contained in $\{z : \mathrm{Re}(z) \geq 0\}$ and contains some points very close to 0. Suppose that one wishes to compute $A^t$ for $0 \leq t \leq 1$. The formula

$$A^t = \mathrm{e}^{t \log(A)}$$

is not recommended because $\log(A)$ may have a very large norm, and it is undefined if 0 is an eigenvalue of $A$. An accuracy of $10^{-8}$ is more than sufficient for plotting the graph of $f(t) := \|A^t\|$, and RADM provides a way of doing this with a minimum of effort. Many other applications of a similar character can easily be devised.

There are four other possible methods of evaluating $A^t$. If $A = I - B$, where $\|B\| \le 1$, then

$$\mathrm{Spec}(A) \subseteq \mathrm{Num}(A) \subseteq \{z : |z-1| \le 1\}.$$

If $s > 0$, then one may define $A^s$ by

$$A^s := \frac{1}{2\pi i} \int_\gamma z^s (zI - A)^{-1} \, \mathrm{d}z,$$

where $\gamma$ is the boundary of the region

$$\{r\mathrm{e}^{i\theta} : 0 < r < 5/2 \text{ and } -3\pi/4 < \theta < 3\pi/4\}.$$

Every point on $\gamma$ except 0 is outside the numerical range of $A$, so the resolvent norm is of order 1 except near 0. The integral is norm convergent for all $s > 0$, but it may develop a singularity at $z = 0$ as $s \to 0+$, so it is not always useful for small $s$.

Alternatively, one might use the expansion

$$A^s = I - \sum_{r=1}^{\infty} c_{r,s} B^r,$$

where

$$c_{r,s} := (-1)^{r+1} s(s-1) \cdots (s-r+1)/r!.$$

The series is norm convergent for all $s \in (0,1)$ because $c_{r,s} \ge 0$ and $\sum_{r=1}^{\infty} c_{r,s} = 1$. However, the convergence of the series is very slow for small $s > 0$, so it is not numerically useful for such $s$. Both of these problems are apparent if 1 is an eigenvalue of $B$, but they also occur if the pseudospectra of $B$ are significant near 1, even if $B$ has no spectrum near 1.

We finally mention that one may also evaluate $A^s$ by using [8] if $s = 1/n$ and by the ODE method of [3].

[illegible] 5.2. We used RADM to compute the $r$th root $C_r$ of the matrix (5.1), with $c := 0.5$, $n = 20$, and $r = 1, \ldots, 10$. Other real powers may be treated in exactly the same way. In the final column of Table 5.2, $C_{r,1}$ and $C_{r,2}$ are two independent computations of $C_r$, both obtained using RADM. The small size of the entries in this column indicates that the results are all reliable to $O(10^{-8})$.

We finally remark that if greater accuracy is needed, then one may use the above procedure to obtain the starting point for a Newton-type iteration.

We used RADM to compute $\|A^t\|$ for the matrix (5.1) with $n = 100$ and $c = 0.6$. We put $t := 2^{-7}r$, where $r$ is a positive integer, $v_1 := \|A^t\|$ computed using RADM, and $v_2 := \|B^r\|$, where $B := A^{1/128}$ is computed by repeated applications of the MATLAB operator `sqrtm`. The two methods give the same answer to within 0.04 for all $t \in (0,2)$, i.e., a relative accuracy of $10^{-4}$. This may seem rather low, but it is more than enough for graph-drawing needs. Both methods computed the norm of $A$ and $A^2$ correctly. Most of the CPU time was used computing the matrix norms, but

TABLE 5.2
*Computation of* $r$*th roots in Example* 5.2.

| $r$ | $\|(C_r)^r - A\|$ | $\|C_r\|$ | $\|C_{r,1}\|/\|C_{r,2}\| - 1$ |
|---|---|---|---|
| 1 | $3.4677e - 007$ | 1.33558842 | $-2.1906e - 009$ |
| 2 | $1.4319e - 007$ | 1.35606917 | $8.9048e - 009$ |
| 3 | $3.8861e - 008$ | 1.57711707 | $4.0503e - 008$ |
| 4 | $1.0367e - 007$ | 1.59766857 | $4.9686e - 008$ |
| 5 | $7.8846e - 008$ | 1.55852362 | $8.1706e - 008$ |
| 6 | $4.6441e - 008$ | 1.50671663 | $2.3817e - 009$ |
| 7 | $8.0153e - 008$ | 1.45657560 | $5.0527e - 008$ |
| 8 | $7.5740e - 008$ | 1.41197706 | $-4.5141e - 008$ |
| 9 | $2.2498e - 007$ | 1.37341090 | $5.3190e - 008$ |
| 10 | $7.8768e - 008$ | 1.34032556 | $1.1992e - 007$ |

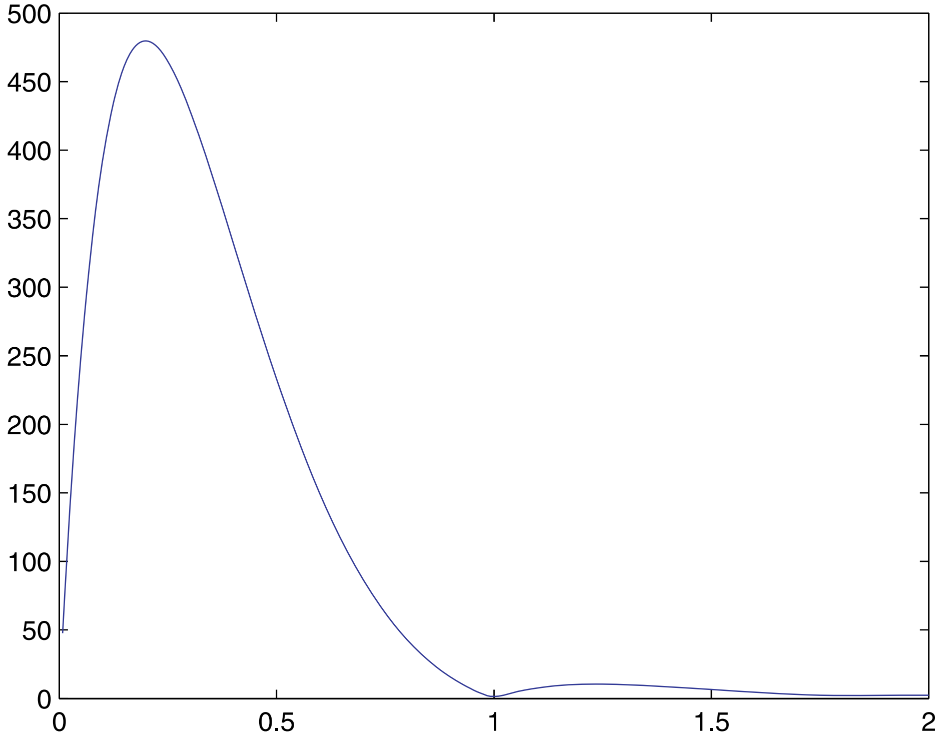


FIG. 5.1. *Graph of* $\|A^t\|$ *for* $0 < t < 2$.

excluding that RADM is substantially faster because it involves one application of `eig` as opposed to seven applications of `sqrtm`. For values of $c$ much smaller than 0.6, neither RADM nor `sqrtm` is accurate. One might also compute $B$ directly using the new algorithm of Guo and Higham [8]. Figure 5.1 shows the graph of the norm and is typical of problems in which pseudospectral behavior is important.

**Acknowledgments.** We should like to thank N. J. Higham, S. Shkarin, E. Shargorodsky, L. N. Trefethen, and the referee for helpful criticism and encouragement.

## REFERENCES

[1] E. B. Davies, *Linear Operators and Their Spectra*, Cambridge University Press, Cambridge, UK, 2007.

[2] P. I. Davies and N. J. Higham, *A Schur–Parlett algorithm for computing matrix functions*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 464–485.

[3] P. I. Davies and N. J. Higham, *Computing* $f(A)b$ *for matrix functions* $f$, in QCD and Numerical Analysis III, Lect. Notes Comput. Sci. Eng. 47, A. Boriçi, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, eds., Springer-Verlag, Berlin, 2005, pp. 15–24.

[4] J. W. Demmel, *The condition number of equivalence transformations that block diagonalize matrix pencils*, SIAM J. Numer. Anal., 20 (1983), pp. 599–610.

[5] S. Friedland, *Normal matrices and the completion problem*, SIAM J. Matrix Anal. Appl., 23 (2002), pp. 896–902.

[6] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.

[7] M. Gu, *Finding well-conditioned similarities to block-diagonalize nonsymmetric matrices is NP-hard*, J. Complexity, 11 (1995), pp. 377–391.

[8] C.-H. Guo and N. J. Higham, *A Schur–Newton method for the matrix pth root and its inverse*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 788–804.

[9] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[10] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.

[11] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra*, Princeton University Press, Princeton, NJ, 2005.

[12] T. G. Wright, *EigTool Software*, available online at http://www.comlab.ox.ac.uk/pseudospectra/eigtool.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1065–1082

# ANALYSIS AND EXPLOITATION OF MATRIX STRUCTURE ARISING IN LINEARIZED OPTICAL TOMOGRAPHIC IMAGING*

DAMON HYDE†, MISHA KILMER‡, DANA H. BROOKS†, AND ERIC MILLER§

**Abstract.** We present a novel method by which the large dense forward matrix **A** involved in a linear inverse diffusion problem can be decomposed into a number of sparse easily computed matrices. We begin by introducing an errorless decomposition which is applicable to a wide array of such imaging problems. Next, we incorporate interpolation into the construction of the matrices to reduce the computational complexity involved in the matrix-vector multiplications necessary to obtain an inverse solution. Error and computational complexity analysis are provided to support these developments. We then present numerical results that illustrate the gain in computational efficiency when the approximation is used in the Tikhonov regularized inverse problem, and show that the use of the approximation has virtually no negative effect on the quality of the reconstructed images. Finally, we discuss applicability to other imaging problems.

**Key words.** structured matrix, matrix approximation, linearized inverse scattering, Tikhonov regularization, image reconstruction

**AMS subject classifications.** 65F10, 65R32, 15A99

**DOI.** 10.1137/060657285

## 1. Introduction.

In diffuse optical tomography (DOT), near-infrared light is introduced to the body from an array of sources on the surface and collected at a number of detectors as it exits [3, 7, 8, 12, 15, 17, 26, 27]. The imaging problem consists of determining images of photon absorption and/or diffusion in the body from this measured photon fluence. In this paper, we consider the problem of efficient image reconstruction from diffuse optical data. Using a linearized model of the relationship between the data and optical absorption coefficient, the specific problem we consider is the efficient solution of the Tikhonov regularized problem:

$$\min_{\mathbf{f}} \|\mathbf{A}\mathbf{f} - \mathbf{g}\|_2^2 + \lambda \|\mathbf{R}\mathbf{f}\|_2^2, \tag{1.1}$$

where the real $N_{data} \times N_{vox}$ matrix **A** is a discretization of a Born-type linearized inverse scattering operator in three dimensions (i.e., the discretization of an integral operator), **f** denotes the vectored form of the absorption image to be determined, and **g** denotes the measured data vector. The regularization term $\lambda\|\mathbf{R}\mathbf{f}\|_2^2$ is necessary to dampen the effects of noise on the quality of the reconstructions as well as to ensure uniqueness of the solution. We employ iterative algorithms as a computationally attractive means to solve (1.1). The nature of these methods is such that the matrix **A** need be utilized only for multiplications of the form $\mathbf{A}\mathbf{x}$ and $\mathbf{A}^T\mathbf{x}$ for an arbitrary

---

*Received by the editors April 14, 2006; accepted for publication (in revised form) by D. P. O'Leary March 27, 2007; published electronically November 2, 2007. This work was funded in part by NSF grants 0139968 and 0208548, and by the Center for Subsurface Sensing and Imaging Systems (CENSSIS).

http://www.siam.org/journals/simax/29-4/65728.html

†Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 (dhyde@ece.neu.edu, brooks@ece.neu.edu).

‡Department of Mathematics, Tufts University, Medford, MA 02155 (misha.kilmer@tufts.edu).

§Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155 (elmiller@ece.tufts.edu).

vector $\mathbf{x}$, and the problem can be solved efficiently for multiple values of $\lambda$ simultaneously. This allows us to concentrate on optimizing for matrix-vector multiplications rather than the more complicated matrix-matrix multiplications.

Though a number of different nonlinear algorithms are currently in use for DOT [6, 7, 8, 29, 25, 27], the associated data sets are generally quite small (on the order of several hundred data points). Technologically, however, sensor systems are rapidly evolving to provide greater spatial density of sources and detectors as well as finer scale sampling in time or frequency. Thus, the size of the data sets available for inversion is increasing far beyond what is typically considered in the nonlinear inversion literature. Scaling these algorithms to larger data sets, such as those considered in this paper, drastically increases the required computational power and makes application of the nonlinear inverse methods either completely infeasible or quite difficult—significant effort and new research would be required in the implementation of these methods on supercomputing-type platforms. Such efforts are indeed well beyond the capabilities of many relevant research and industrial organizations (e.g., Advanced Research Technologies (ART), who provided us with the phantom data used in this paper). In such settings the linear solution is the only feasible approach to the problem. Thus, though nonlinear methods for DOT are certainly under consideration, there remains relevance to considering the linear form of the problem as well.

Toward this end, we point to recent research using linear methods for DOT and related areas where the results presented here may prove relevant [4, 9, 14, 16, 18, 19, 30]. Moreover, we emphasize that the methods we present in this paper are really only loosely dependent upon the specific form of the Green's functions. As long as (a) the Green's functions are smooth and display some level of spatial invariance and (b) the data collection scheme is also regular, the methods detailed here will be usable. This makes our work relevant to a wide range of problems, of which DOT is but a single application. For example, the ideas here could be applicable to any large scale diffusive-type inverse problem (where data set size and voxel dimensionality prevent the use of nonlinear methods) such as diffusive-type electromagnetic induction imaging arising in geophysical applications, photothermal/photoacoustic nondestructive evaluation, bioluminescence tomography (BLT), and fluorescence molecular tomography (FMT). As a specific example, take FMT [21, 20]. When imaging fluorescence, the collected fluorescence data are approximately linearly related to the image, and using methods such as the normalized Born ratio, it is possible to minimize the effects of inhomogeneities in the background optical parameters [28]. Moreover, for this problem, the development of CCD detectors and tomographic data acquisition systems is leading to imaging problems even larger than those considered here. Thus we would anticipate that large computational gains could be achieved using the method in this paper.

The difficulty in practice with solving the Tikhonov problem is that the matrix $\mathbf{A}$ is dense and extremely large, with $N_{data}$ and $N_{vox}$ being on the order of $10^4$–$10^5$. For this work, we restricted ourselves to working with a system configuration consisting of a slab transmission geometry using time domain data collection. In this configuration, the region to be imaged is modeled as the volume contained between two parallel infinite planes. The solution volume is then a compact finite section of this infinite volume [12, 15]. Sources are located along one plane, directed into the volume, while detectors are arrayed along the other plane to collect exiting light. At each source location a picosecond laser is pulsed, and the time dependent intensity is recorded at a number of detectors for time gates of several nanoseconds. Given that each detector may collect hundreds or even thousands of time points for each source pulse, the total data set rapidly becomes extremely large. In systems of this sort, it is not uncommon to see hundreds of thousands of data points collected in a single imaging session. Thus, even with a very small number of voxels, the size of **A** rapidly becomes prohibitive. Storing such a matrix in double precision can require gigabytes of storage space and thereby renders useless any algorithm requiring the up-front computation of **A**.

Therefore, our primary goal is to represent the full matrix **A** such that both the time required for its computation and the necessary storage space are significantly reduced. In particular, we look to exploit the structure of the matrix **A** so that redundant matrix entries are not explicitly computed/stored. These redundancies arise from a combination of the regularity in the data acquisition process and the structure of the Green's functions used to compute the matrix elements. As we will show, only a relatively small amount of information is needed to implicitly represent every matrix entry, so matrix-vector products can be performed on-the-fly simply by reference to a particular source-detector pattern and the small amount of stored information. However, even with a more compact representation, a significant amount of time is still required to evaluate matrix-vector products involving these matrices. Given that these products may need to be evaluated numerous times to obtain a solution, a secondary goal of this work was the reduction of the computational complexity involved in executing matrix-vector products. To do this, we effectively replace the implicit representation of **A** by an approximation that can be applied to vectors more quickly without degradation of the reconstructed images.

To achieve the first goal, we take into account the spatial invariance of the integral equation kernel from which **A** is derived. Using a change of variables, we are able to exploit the matrix redundancies more readily. This allows us to represent **A** in terms of the product of a single small matrix and a collection of sparse, easily computed matrices. This decomposition is made possible by a regular sampling pattern and planar shift invariance in the kernel.

To achieve the second goal, we introduce an interpolation approach, applied in the aforementioned coordinate system, to further reduce the number of matrix components that must be explicitly computed. The utility of the interpolation approach arises from the smoothness of the kernel of the integral equation from which **A** is derived. In turn, this suggests applicability to other diffusing imaging problems with similarly smooth kernels. By choosing an interpolation method which is expressible in matrix form, we are then able to achieve reduction in the amount of computation required to implement the matrix vector products $\mathbf{Ax}$ and $\mathbf{A}^T\mathbf{x}$. These gains are shown to be directly proportional to the number of nodes used in the interpolation scheme, allowing for a direct tradeoff between computation time and accuracy.

Applying these two steps to the overall problem at hand, we are able to obtain a dramatic decrease in the amount of time required to obtain a solution to the problem (1.1) using an iterative algorithm. We present several sets of numerical results, both with and without the use of interpolation, to approximate **A**. Using phantom simulations, we show that while the two solutions are not identical, visually they are very similar, and mathematically they have effectively the same mean squared error with respect to the true image.

This paper is organized as follows. In section 2, we provide an overview of the physics and mathematics behind the construction of the matrix **A**. In section 3.1, we show how to represent the matrix in compact form. Section 3.2 is devoted to presenting an interpolation-based approximation to the matrix. Section 4 details the precise reductions in computational complexity. Inversion results using the algorithm in [13] on a simulated data set are presented in section 5 along with an analysis of the error introduced by the interpolation. Finally, in section 6, we summarize our results and outline potential further extensions.

**2. Problem description.** The diffusion approximation model arises as an approximation to the radiative transport equation [5], and takes the form [2]

$$\left(\nabla \cdot \gamma^2 \nabla - \mu_a - \frac{1}{v}\frac{\partial}{\partial t}\right)\Phi(\mathbf{r},t) = -q(\mathbf{r},t), \qquad \gamma^2 = \frac{1}{3[\mu_a + (1-\bar{p})\mu_s]}. \tag{2.1}$$

Here, $\Phi(\mathbf{r},t)$ represents the photon density at a location $\mathbf{r}$ and time $t$. Sources are represented by $q(\mathbf{r},t)$. The two physical parameters of interest are $\mu_a$ and $\mu_s$, the absorption and scattering parameters, respectively. Additionally, $v$ is the speed of light in the medium, and $\bar{p}$ is the mean cosine of the scattering angle.

The goal in this work is to recover $\mu_a$, assuming that $\mu_s$ is constant, given knowledge of the sources. Since it is clearly a nonlinear problem to recover $\mu_a$ from (2.1), the equation is frequently linearized by assuming that the overall system is approximately homogeneous. One can then use Green's functions for the homogeneous case to reformulate the imaging problem as one of finding the perturbation about some known background absorption level. Therefore, we let $\mu_a$, $\mu_s$ denote the known background values of absorption and scattering, and we use $\eta(\mathbf{r}')$ to denote the unknown perturbations of absorption about the known background value.

Assuming that the source term $q(\mathbf{r},t)$ is a delta function located at position $\mathbf{r}$ and time $t$, a solution $\Phi$ in the form of a Green's function can be derived. For the slab transmission geometry that we consider in this paper, the two-point time domain Green's function is [1]

$$\begin{aligned} g_{slab}^{(\Phi)}(\mathbf{r},\mathbf{r}',t,t_0) = {} & \frac{\exp\left\{-\left[\mu_a c(t-t_0) + \frac{d^2}{4\gamma^2(t-t_0)}\right]\right\}}{[4\pi\gamma^2(t-t_0)]^{3/2}} \\ & \times \sum_{n=-\infty}^{\infty}\left[\exp\left(\frac{-(z-2z_d n - z_0)^2}{4\gamma^2(t-t_0)}\right)\right. \\ & \left. - \exp\left(\frac{-(z-2z_d n + z_0)^2}{4\gamma^2(t-t_0)}\right)\right], \\ & d = \sqrt{x^2+y^2}, \quad \text{where } \mathbf{r}-\mathbf{r}' = (x,y,z), \\ & z_o = [(1-\bar{p})\mu_s]^{-1}. \end{aligned} \tag{2.2}$$

This equation models the transmission of light from point $\mathbf{r}$, leaving at time $t_0$, and arriving at point $\mathbf{r}'$ at time $t$. The constant $z_d$ represents the thickness of the slab in question and is used to generate the multiple image sources needed to satisfy the boundary conditions of the system [11, 24]. The placement of these image sources results in the Green's function taking a value of zero at the boundary of the diffusive medium. Finally, the distance $z_0$ represents the source depth; because we use a slab geometry model, all sources are located at this height. It is presumed that all light from the source travels a short distance into the medium before proceeding to scatter randomly. This is modeled by assuming the sources to be isotropic and placing them one mean scattering length into the medium.

These Green's functions are used in a model of the sensing system based on the first order Born approximation [22]. This approximation assumes that the total received signal is the sum of the signal for a homogeneous system and a perturbation due to $\eta(\mathbf{r}')$, the inhomogeneities in $\mu_a$:

$$\Gamma_{total} = \Gamma_{homog} + \Delta\Gamma(\eta(\mathbf{r}')). \tag{2.3}$$

Using the background optical properties, $\Gamma_{homog}$ can be computed and subtracted from $\Gamma_{total}$, leaving $\Delta\Gamma$. We now concentrate on obtaining a description of $\Delta\Gamma$ as a linear function of $\eta(\mathbf{r}')$.

Under the first order Born approximation, $\Delta\Gamma$ is dependent only on first order scattering; thus by integrating across $\Omega$, the volume to be imaged, an equation for $\Omega$ can be written as

$$\Delta\Gamma(\mathbf{s},\mathbf{d},t,t_0) \approx -\int_\Omega \int_{-\infty}^{\infty} [g^{(\Gamma)}_{slab}(\mathbf{s},\mathbf{r}',t',t_0)\eta(\mathbf{r}')g^{(\Phi)}_{slab}(\mathbf{r}',\mathbf{d},t-t',t_0)]dt'd\mathbf{r}'. \tag{2.4}$$

Here, $\triangle\Gamma(\mathbf{s},\mathbf{d},t,t_o)$, the change in the photon fluence measured at location $\mathbf{d}$ at time $t$ due to inhomogeneities in the background absorption for a source at location $\mathbf{s}$, is equal to the integral of all first order scattering throughout the volume. The Green's function $g^{(\Gamma)}_{slab}(\mathbf{r},\mathbf{r}',t,t_0)$ is the spatial gradient of $g^{(\Phi)}_{slab}(\mathbf{r},\mathbf{r}',t,t_0)$ with respect to a unit normal extending out of the solution volume. This gives $g^{(\Gamma)}_{slab}(\mathbf{r},\mathbf{r}',t,t_0)$ the form

$$\begin{aligned} g^{(\Gamma)}_{slab}(\mathbf{r},\mathbf{r}',t,t_0) &= \frac{\exp\left\{-\left[\mu_a c(t-t_0)+\frac{d^2}{4\gamma^2(t-t_0)}\right]\right\}}{[4\pi\gamma^2(t-t_0)]^{3/2}} \\ &\quad\times\sum_{n=-\infty}^{\infty}\left[\frac{-2\,(z-2z_d n-z_0)}{4\gamma^2(t-t_0)}\exp\left(\frac{-(z-2z_d n-z_0)^2}{4\gamma^2(t-t_0)}\right)\right. \\ &\qquad\left.+\frac{2(z-2z_d n+z_0)}{4\gamma^2(t-t_0)}\exp\left(\frac{-(z-2z_d n+z_0)^2}{4\gamma^2(t-t_0)}\right)\right], \\ d &= \sqrt{x^2+y^2}, \quad \text{where } \mathbf{r}-\mathbf{r}'=(x,y,z), \\ z_o &= [(1-\bar{\rho})\mu_s]^{-1}. \end{aligned} \tag{2.5}$$

This relationship is necessary, as the photon density is not a directly measurable quantity. By taking the gradient of the photon density, we obtain the photon fluence, the intensity of the light exiting from the boundary at the location of the detector. This fluence is a quantity which we are capable of measuring with detectors placed on the surface.

Because the system is causal, (2.2) is zero for $t < t_0$. Additionally, presuming that the timescale can be adjusted such that $t_0 = 0$, the second integral in (2.4) will have support only for $t'$ such that $0 \le t' \le t$, and the dependence upon $t_0$ can be dropped from (2.4). Discretizing (2.4) in piecewise constant fashion for each voxel converts the spatial integration into a summation. Combining these modifications results in

$$\triangle\Gamma(\mathbf{s},\mathbf{d},t) \approx -\sum_{i=1}^{Nvox} dV_i \int_0^t [g^{(\Gamma)}_{slab}(\mathbf{s},\mathbf{r}'_i,t')g^{(\Phi)}_{slab}(\mathbf{r}'_i,\mathbf{d},t-t')\eta(\mathbf{r}'_i)]dt', \tag{2.6}$$

where $dV_i$ is the volume of the $i$th voxel and the $r'_i$'s are locations of voxel centers. For simplicity and maximum computational gain, we assume that $dV_i$ is constant for all voxels and simply note it as $dV$. The above equation then serves as a basis from which to construct the discrete linear model $\mathbf{Af} \approx \mathbf{g}$, where $\mathbf{f}$ is the vector of unknown absorption values at each of the voxels in the image. This equation is ill-posed in the sense that a least-squares solution to the system would be hopelessly contaminated by noise. Therefore, we solve instead the Tikhonov regularized problem (1.1).

From (2.6), we see that the entry in the matrix $\mathbf{A}$ associated with voxel (column) $i$ and row corresponding to source $\mathbf{s}$ and detector $\mathbf{d}$ at time $t$ is

$$J_i^{(\Gamma)}(\mathbf{s}, \mathbf{d}, t) \approx -dV \left( \sum_{j=1}^{T_t} w_j \left[ g_{slab}^{(\Gamma)} \left(\mathbf{s}, \mathbf{r}'_i, t'_j\right) g_{slab}^{(\Phi)} \left(\mathbf{r}'_i, \mathbf{d}, t - t'_j\right) \right] \right), \tag{2.7}$$

where in this case the approximation notation conveys the fact that the integral was evaluated numerically using the composite trapezoid rule on a regular grid, and the $w_j$ denote the weights of the composite trapezoid rule.

As mentioned in the introduction, it is not feasible to naively construct and store each entry in $\mathbf{A}$. However, it is clear from the above Green's functions that in a slab geometry there is some degree of spatial invariance in the kernels. In the following section, we describe how to utilize the invariance to store only a minimum of information to represent every entry in $\mathbf{A}$, and to utilize the stored information to perform the matrix-vector products necessary to employ an iterative algorithm for solving (1.1).

## 3. Exploiting matrix structure.

There is a significant amount of redundancy in the forward matrix. By eliminating the excesses involved in computing the same value multiple times, we can reduce the time required to generate the matrix. We are able to store each computed value only once and reuse it as needed. This reuse takes the guise of a series of selection matrices: extremely sparse, easily formed matrices consisting entirely of ones and zeros.

### 3.1. Change of coordinates.

In (2.2), (2.5), and (2.7), the X-Y coordinates of the source-voxel and detector-voxel differences enter into the equation only as radial distances $\sqrt{x^2 + y^2}$. Because of this, the absolute X-Y locations involved are irrelevant to the computation, and it is possible to change from the original absolute Cartesian coordinates to a different coordinate system based on these radial distances. Equation (2.7) expresses each matrix component as the convolution of two Green's functions with respect to time. Given the convolution involved in obtaining the matrix components, this new coordinate system can be seen as two joined cylindrical coordinate systems, with the central axis of one lying upon the radial boundary of the other. To see this, let $(X_s, Y_s, Z_s)$ represent the absolute location of the source and $(X_r, Y_r, Z_r)$ and $(X_v, Y_v, Z_v)$ represent the absolute locations for the detector and voxel, respectively. Define the new variables:

$$\begin{aligned} D_1 &= ((X_v - X_s)^2 + (Y_v - Y_s)^2)^{1/2}, \\ Z_1 &= Z_v - Z_s, \\ D_2 &= ((X_v - X_r)^2 + (Y_v - Y_r)^2)^{1/2}, \\ Z_2 &= Z_r - Z_v. \end{aligned} \tag{3.1}$$

These four values are sufficient for computing the value of (2.7), regardless of the absolute position of the three initial sets of $(x, y, z)$ coordinates. Further, because we are modeling a slab geometry, the $z$-coordinates for the sources are fixed and known,

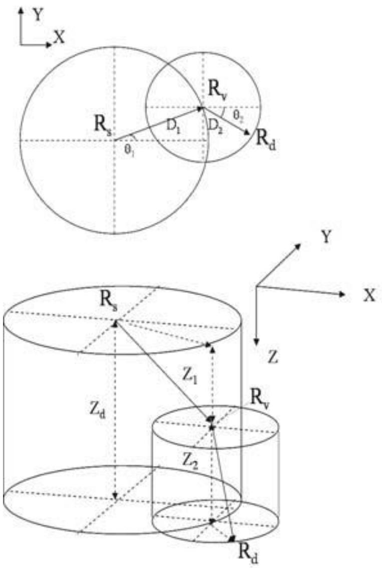


FIG. 3.1. *Dual cylindrical coordinate system. This illustration depicts visually the spatial invariance of* (2.7)*, which expresses the matrix values as the numerical convolution of two Green's functions. Here, the source is located at $R_s$, the center of the top of the larger cylinder; the voxel under consideration is at $R_v$, the top center of the smaller cylinder; and the detector is located at $R_d$, a point on the lower rim of the cylinder. Given this arrangement, the radial location of the small cylinder with respect to the larger, and the radial location of the detector with respect to the small cylinder, can both be changed arbitrarily without affecting the resulting value of* (2.7)*. Additionally, given a slab geometry of fixed thickness $Z_d$, the entire system is shift invariant to changes along the X-Y plane.*

as are the $z$-coordinates for the detectors. Therefore, the only time that $Z_1$ and $Z_2$ change is when $Z_v$ changes, and thus only one of $Z_1, Z_2, Z_v$ is needed in order to compute the other two. As the locations of the source, detector, and voxel are allowed to vary, these changes will be reflected in changes to the triple $(D_1, D_2, Z_v)$. This dual cylindrical coordinate system is shown in Figure 3.1. Note that the two radial angles $\theta_1$ and $\theta_2$ do not appear in (3.1). Clearly, (2.7) is independent of $\theta_1$ and $\theta_2$ and is therefore invariant to changes in the angles. Therefore, given fixed $t$, for each source-voxel/detector-voxel pairing in XYZ space which maps to the same $(D_1, D_2, Z_v)$ triple, the corresponding matrix entry will be the same. Note that this means that all sets of three points with the same voxel height and the same length x-y projections of the source-voxel and voxel-detector distances require identical computations.

Using this new coordinate system, we now return to the original problem, with all of the source and detector positions, and examine those positions within this dual-cylindrical system. In practice, many source-detector configurations fall into one of two categories: fixed array or raster scanned. For the fixed array case, two grids are defined, one for the sources and one for the detectors. For each source location, data are collected at all of the detectors. In a raster scanned system, a source grid is defined, along with a number of detector locations, fixed relative to the source. In both cases, high levels of redundancy in the $(D_1, D_2, Z)$ triplets will be present. This means that a large number of repeated operations are performed if each component of the matrix $\mathbf{A}$ is explicitly computed. Table 1 shows redundancy levels for several common source-detector configurations based on a raster scan and two uniform grids

Table 1

*Computational/storage improvements for a variety of source-detector configurations. This table reflects the amount of redundancy present for three examples of common source-detector geometries. The reduction is presented as the ratio of total to unique number of matrix elements. Thus, for the second example given, explicitly computing every matrix element results in each unique computation being done an average of* 690 *times. By eliminating these excess computations and simply computing each unique element once, we are able to significantly reduce the amount of time required to compute the forward matrix.*

| SD configuration | Computational/storage reduction |
|---|---|
| 7 × 7 Raster scanned | 49× |
| 5 × 5 Sources over<br>5 × 5 Detectors<br>10 × 10 × 10 Voxels | 690× |
| 10 × 10 Sources over<br>10 × 10 Detectors<br>10 × 10 × 10 Voxels | 3844× |

Table 2

*Summary of index notation. Note that* $N_{pts} \leq N_{vox} N_D N_S$.

| Symbol | Meaning |
|---|---|
| $N_z$ | No. grid pts in z-direction |
| $N_{vox}$ | No. of voxels |
| $N_{pts}$ | No. of unique $(D_1, D_2, Z)$ triples |
| $N_S$ | No. of sources |
| $N_D$ | No. of detectors |
| $N_t$ | No. of time pts |
| $N_{comp}$ | No. of interp. nodes in 3-space |

of sources and detectors.

Now we are ready to consider taking advantage of this redundancy to represent the matrix **A** (see Table 2 for definitions of the dimension notation). The matrix **A** has $N_t N_D N_S$ rows and $N_{vox}$ columns. We order the rows of **A** such that the inner loop is over time, then detectors, then sources. It will be convenient to consider the structure of $\mathbf{A}^T$ instead of **A**. Using $\mathbf{A}_{ij}$ to denote the $N_{vox} \times N_t$ submatrix holding the entries given by (2.7) for the $i$th source and the $j$th detector, the matrix $\mathbf{A}^T$ has the block structure

$$\mathbf{A}^T = [\mathbf{A}_{11} \quad \dots \quad \mathbf{A}_{1N_D} \quad \mathbf{A}_{21} \quad \dots \quad \mathbf{A}_{N_S N_D}]. \tag{3.2}$$

However, given the redundancy noted above, explicit computation of each $A_{ij}$ is unnecessary. Let us assume that there are $N_{pts}$ unique $(D_1, D_2, Z)$ coordinate triplets, given all source-voxel-detector combinations. Clearly, $N_{pts} \leq N_{vox} N_D N_S$. Each of these triplets is encountered again at each time step, for a total of $N_{pts} N_t$ unique evaluations of (2.7). Let $\mathbf{A}_s$ denote the $N_{pts} \times N_t$ matrix containing these values. Thus instead of computing and storing all $N_{vox} N_D N_S N_t$ entries in **A**, we will only need to evaluate and store the $N_{pts} N_t \leq N_{vox} N_D N_S N_t$ unique entries of $\mathbf{A}_s$.

We can represent **A** in terms of $\mathbf{A}_s$ using a series of selection matrices of size $N_{vox} \times N_{pts}$, where each row consists of all zeros, except for a single "1" to select the appropriate row from $\mathbf{A}_s$. Placing these selection matrices into the previous expression for **A** results in

$$\mathbf{A}^T = [\mathbf{S}_{11}\mathbf{A}_s \quad \dots \quad \mathbf{S}_{1N_D}\mathbf{A}_s \quad \mathbf{S}_{21}\mathbf{A}_s \quad \dots \quad \mathbf{S}_{N_S N_D}\mathbf{A}_s]. \tag{3.3}$$

It is possible to rewrite (3.3) in a form which exploits the underlying Kronecker structure of the matrix:

$$\mathbf{A}^T = [\mathbf{S}_{11} \quad \dots \quad \mathbf{S}_{1N_D} \quad \mathbf{S}_{21} \quad \dots \quad \mathbf{S}_{N_S N_D}][\mathbf{I}_{N_S * N_D} \otimes \mathbf{A}_s]. \tag{3.4}$$

We now have a compact representation of the matrix $\mathbf{A}$ that may be used inside the iterative solver to produce the necessary matrix-vector products $\mathbf{A}x$ or $\mathbf{A}^T x$. All entries of the matrix $\mathbf{A}$ need not be explicitly formed.

Furthermore, there is structure present within the selection matrices themselves. If $\mathbf{A}_s$ is arranged such that it has block structure,

$$\mathbf{A}_s = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \\ \vdots \\ \mathbf{Z}_{N_z} \end{bmatrix}, \tag{3.5}$$

where each block corresponds to the set of $(D_1, D_2, Z)$ triplets sharing a particular $Z$ value, the selection matrices themselves have a Kronecker structure. This is because, assuming a uniform grid for the voxels, the same set of $(D_1, D_2)$ values will be required from each Z-slice. Thus we have

$$\mathbf{S}_{ij} = \mathbf{I}_{N_z} \otimes \tilde{\mathbf{S}}_{ij}, \tag{3.6}$$

where the matrix $\tilde{\mathbf{S}}_{ij}$ is the selection matrix required to extract the appropriate rows from any of the $Z_k$ matrices.

**3.2. Interpolation.** While this change-of-coordinates representation provides a significant reduction in the amount of overhead required to compute and store the matrix $\mathbf{A}$, it does not provide any gains when that matrix is used in matrix-vector multiplications. While operations on $\mathbf{A}$ can be done block by block, the overall size of $\mathbf{A}$ is still exceedingly large. When used in an iterative scheme where multiple matrix-vector products are required for each iteration, the time involved in each product becomes a limiting factor. The desire to accelerate these products, as well as further reduce the required initial computation, motivates the next step in our method.

Recall that even though we have reduced the number of distinct entries that need to be computed to represent $\mathbf{A}$, each of these distinct entries requires the evaluation of the expression (2.7). These are clearly expensive to compute because of the numerous evaluations of the Green's functions and multiple summations. It is this function evaluation whose explicit calculation we hope to minimize. Therefore, we propose to use interpolation to aid in the function evaluation. Not only does this reduce the overall amount of initial computation required to approximate each matrix entry, but as we will illustrate shortly, it has the added benefit of speeding up matrix-vector products.

We utilized interpolation expressible in the linear form

$$\mathbf{A}_s \approx \mathbf{Q}\mathbf{V}. \tag{3.7}$$

Here, $\mathbf{Q}$ is of size $N_{pts} \times N_{comp}$ and is the interpolation matrix, while $\mathbf{V}$ is $N_{comp} \times N_t$, consisting of the smaller set of values which must be explicitly computed. $N_{comp}$ is the number of nodes to be computed for use in the interpolation scheme and is chosen to be significantly smaller than $N_{pts}$. While all further results, including computational complexities, will be shown with respect to the specific linear interpolation scheme

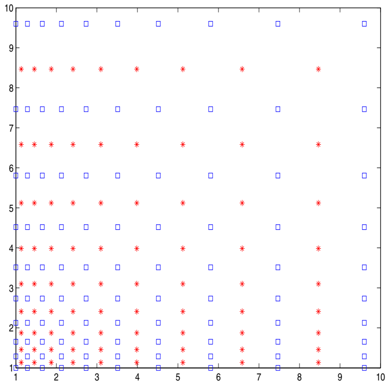


FIG. 3.2. *Graphical description of the sampling technique used with the interpolation. The sets of red and blue points (asterisks and squares, respectively) show the two grids which were used to sample $D_1 - D_2$ slices of the sample space. For each Z-value, one of the two grids was selected, alternating as the Z-value was stepped from one value to the next.*

we chose to use, this process could be used with any interpolation method that can be expressed in matrix format.

Evaluation of the function in (2.7) will occur at a set of $N_{comp}$ interpolation nodes in $(D_1, D_2, Z)$ space at each of the $N_t$ values $\{t_1, \ldots, t_{N_t}\}$. The linear interpolation we use is based on a Delaunay tessellation of the $(D_1, D_2, Z)$ space. This tessellation uses the interpolation nodes as vertices of a tetrahedral mesh. To determine the value at each point on the more dense grid to which we interpolate, we first determine inside which tetrahedron the point lies. Barycentric coordinates of the desired point are then computed with respect to the vertices of the enclosing tetrahedron, and those coordinates are used as the weights of the interpolation. For example, if the coordinates of the four interpolation nodes comprising the encircling tetrahedron are given by $(D_1^{(i)}, D_2^{(i)}, Z^{(i)})$, $i = [1, \ldots, 4]$, with corresponding function values given by $v_i$, $i = [1, \ldots, 4]$, then the barycentric coordinates of the desired point $(D_1, D_2, Z)$ can be determined as

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} D_1^{(1)} & D_1^{(2)} & D_1^{(3)} & D_1^{(4)} \\ D_2^{(1)} & D_2^{(2)} & D_2^{(3)} & D_2^{(4)} \\ Z^{(1)} & Z^{(2)} & Z^{(3)} & Z^{(4)} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} D_1 \\ D_2 \\ Z \\ 1 \end{bmatrix}. \tag{3.8}$$

These barycentric coordinates become the weights of the interpolation, to give an approximate value $v$ at the point $(D_1, D_2, Z)$ of

$$v = a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4. \tag{3.9}$$

Arranging the weights in $\mathbf{Q}$ and the coarse grid values in $\mathbf{V}$, the $N_{pts} \times N_{comp}$ matrix $\mathbf{Q}$ will be sparse with only four nonzero entries per row, while $\mathbf{V}$ will be dense. It remains to decide how to choose the position of the interpolation nodes.

Looking again at (2.2), there is an exponential relationship between the calculated value and the values of $D_1$ and $D_2$. Because of this, a linear sampling method is unlikely to give acceptable results when combined with a linear interpolation method. This is especially true when $D_1$ and $D_2$ are close to zero, where the value of (2.2)

Table 3

*Number of floating point operations for various matrix-vector products. This assumes that multiplication by 1 can be done at no cost. The vector $w$ is the component of $v$ corresponding to the data points from a single source-detector pair. Note that the use of $A_s$ alone does not provide computational gains when computing matrix-vector products. It is the combination of the selection and interpolation matrices (the $S_{ij}Q$ product) and the initial $Vw$ multiplication that provide the reduction in total required computation.*

| Product | Flops (in big-Oh) |
|---|---|
| $A^T v$, $A$ dense | $O(N_D N_S N_t N_{vox})$ |
| $A_s w$ | $O(N_{pts} \times N_t)$ |
| $A^T v = S(I_{N_D N_S} \otimes A_s)v$ | $O(N_D N_S N_t N_{vox})$ |
| $(S_{ij}Q)(Vw)$ | $O(4N_{vox} + N_{comp}N_t)$ |
| Approx. $A$ using $A_s w \approx QVw$ | $O(N_D N_S (4N_{vox} + N_{comp}N_t))$ |

is changing very rapidly. Initial experimentation confirmed that this held true in practice. As an alternative sampling method, a grid with exponential spacing in $D_1$ and $D_2$ was developed, as illustrated in Figure 3.2. By clustering a larger number of sample points near the origin, it was possible to achieve significantly better interpolation results for the same values of $N_{pts}$, with acceptable error levels in terms of the quality of the resulting reconstructions. To efficiently sample the space, two grids, red and blue, were established, with the nodes of one grid centered between the nodes of the other. We alternated between the two grids as we stepped down the Z-axis, as indicated by the red and blue grids in Figure 3.2. In order to ensure that the grid covered all of the necessary space, the final slice always used the blue grid, even if this meant that two adjacent slices utilized the same grid.

**4. Computational complexity.** To this point, we have been primarily concerned with the up-front costs in terms of storage and computation time for representing the matrix. We decreased the storage for the matrix by exploiting redundancy. With the interpolation approximation, we also reduced the initial time needed to represent entries in the (approximate) matrix, and the storage requirement for $\mathbf{A}_s$ has decreased from $N_{pts}N_t$ to $4N_{pts} + N_{comp}N_t$ double values stored in memory. Further, the addition of the interpolation step adds the potential to decrease the overall computational complexity of executing each matrix-vector product involved in solving the minimization problem. Because the computations in the matrix-vector product that correspond to individual source-detector pairs are independent, we will simply examine the computation required for a single source-detector pair.

As a benchmark, the number of floating point operations (i.e., multiplications and additions) required for matrix-vector products using the dense formulation and also the formulation in section 3.1 are given in Table 3. Note that the flop count is not reduced over the dense formulation when using the sparse representation in section 3.1. (Note: multiplication by the selection matrix requires no flops.)

The situation changes when utilizing interpolation. At each source-detector pair, the equation to be evaluated is

$$\mathbf{S}_{ij}\mathbf{QVx}. \tag{4.1}$$

Here, the first step is to combine the selection and interpolation matrices. Again, the product of these two matrices can be computed at no cost. The reason for taking this "product" first is that in general $N_{vox} < N_{pts}$, and thus the resulting sparse matrix will have only $4N_{vox}$ nonzeros as opposed to the $4N_{pts}$ nonzeros that are in $Q$.

Therefore, we compute $S_{ij}QVw$ in three steps:

Table 4

*The various matrices used in our approximate forward model, their sizes, and their number of nonzero values. $A_s$ is the small version of the forward matrix containing only the unique matrix elements. $S_{ij}$ are the selection matrices used to retrieve the blocks of the full matrix from $A_s$. $A$ is the full weight matrix. $Q$ is the matrix responsible for implementing the linear interpolation, and $V$ is a small dense matrix of unique values such that $A_s \simeq QV$. For descriptions of other variables used, see Table 2.*

| Matrix | Dimensions | Nonzeros |
|---|---|---|
| $A_s$ | $N_{pts} \times N_t$ | $N_{pts}N_t$ |
| $S_{ij} = I_{N_z} \otimes \tilde{S}_{ij}$ | $N_{vox} \times N_{pts}$ | $N_{vox}$ |
| $A$ | $N_D N_S N_t \times N_{vox}$ | (Sparse rep.) $N_D N_S N_{vox} + N_{pts}N_t$ |
| $Q$ | $N_{pts} \times N_{comp}$ | $4N_{pts}$ |
| $V$ | $N_{comp} \times N_t$ | $N_{comp}N_t$ |
| Approx. $A$ | | $N_D N_S N_{vox} + 4N_{pts} + N_{comp}N_t$ |

- Form the product $W_{ij} = S_{ij}Q$.
- Compute the matrix-vector product $Vw$.
- Compute the matrix-vector product $W_{ij}(Vw)$.

The first step is "free." The second step requires $O(N_{comp}N_t)$ flops. Since $W_{ij}$ is $N_{vox} \times N_{comp}$ but has only four nonzero entries per row, the product $W_{ij}(Vw)$ requires an additional $O(4N_{vox})$ flops. Thus, the cost of the product $S_{ij}QVw$ is $O(4N_{vox} + N_{comp}N_t)$ flops. There is one such product for every source-detector pair, and therefore products with the approximation to $A$ formed by using $A_s \approx QV$ cost $O(N_D N_S(4N_{vox} + N_{comp}N_t))$ flops. Comparing this to the total number of flops required for the dense formulation, we can see that the reduction in required computation is dependent upon the number of interpolation nodes $N_{comp}$ and the total number of voxels $N_{vox}$. The computational costs and storage requirements for the various steps are detailed in Tables 3 and 4.

**5. Simulation results.** For the regularization matrix $\mathbf{R}$, a first order approximation to the gradient was utilized, generated as

$$\mathbf{R} = \begin{bmatrix} R_x \\ R_y \\ R_z \end{bmatrix}, \tag{5.1}$$

where each of the three submatrices are discrete approximations to the first order derivative along the associated axis.

Rather than simply run an iterative algorithm such as LSQR [23] with an augmented matrix once for each trial value of $\lambda$ in (1.1), we used the algorithm in [13] so that the results could be run simultaneously on an array of $\lambda$'s. The number of iterations was fixed at fifty, chosen by first running the algorithm without any regularization and performing an L-curve analysis [10] across the iterations. It is reasonable to presume that a given regularized system should have sufficiently converged by several iterations past the corner of the L-curve for the unregularized problem. For the regularized problem, selection of the appropriate regularization parameter was done through the use of an L-curve analysis at the final iteration.

After some experimentation, it was found that for our first data set, where the ground truth was known, the minimum error solution was consistently at a point which would be considered underregularized according to the L-curve. Visual analysis of a second data set, provided by ART, suggested that a similar situation existed with that data. As such, the results shown for the known phantom are those at the

Table 5

*Interpolation levels, with corresponding number of computed nodes and the resulting error induced in the solution for data set* 1. *Error is computed as* $\frac{\|Computed-Actual\|_2}{\|Actual\|_2}$. *The interpolation level noted in column* 1 *denotes the initial number of interpolation nodes along each dimension of the space. To eliminate unnecessary evaluations of* (2.7), *only those nodes needed to approximate* $A_s$ *were computed.*

| Interpolation level | Number of computed points | Induced error |
|---|---|---|
| None | 11760 | 0.7336 |
| (40,40,40) | 7188 | 0.7344 |
| (30,30,30) | 4659 | 0.7352 |
| (20,20,20) | 2109 | 0.7362 |
| (15,15,15) | 1098 | 0.7409 |
| (10,10,15) | 578 | 0.7294 |
| (10,10,10) | 402 | 0.7404 |

minimum error point, while those shown for the second data set were chosen to be "underregularized" by a similar order of magnitude.

Of course, critical to determining the utility of the interpolation is evaluation of the error introduced into the reconstructions. Interpolation levels are denoted in what follows as a triplet (a,b,c), where the values define the number of grid points along each of the $(D_1, D_2, Z)$ axes, respectively. However, the rectangular grid computed using the values will contain some nodes which will not be needed during the interpolation step. Rather than compute their values and not use them, we simply eliminate these nodes. The numbers in the second column of Table 5 give the number of nodes remaining after this elimination.

We show results for two simulated data sets, each using a number of different interpolation levels. For our first data set, where ground truth was known, we present reconstruction images for two interpolation levels as well as the fully computed result. Additionally, we report error levels for a further four interpolation levels. For our second data set we present three reconstructions, two using interpolation and one without, and give an analysis of relative error levels.

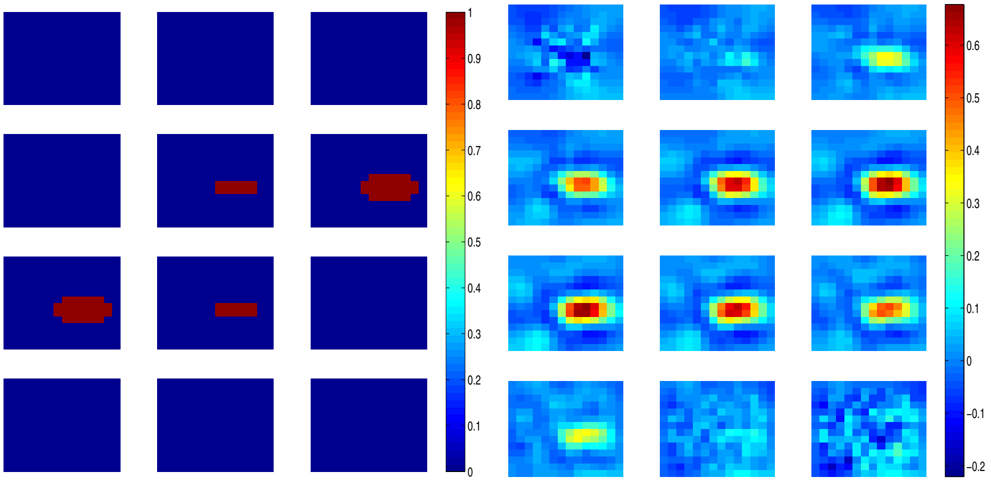
For each data set, relative error levels are computed with respect to the other reconstructions and with respect to ground truth in the case it is known. All error levels are computed as

$$E_{\mathbf{f}_b}(\mathbf{f}_a) = \frac{\|\mathbf{f}_b - \mathbf{f}_a\|_2}{\|\mathbf{f}_b\|_2}, \tag{5.2}$$

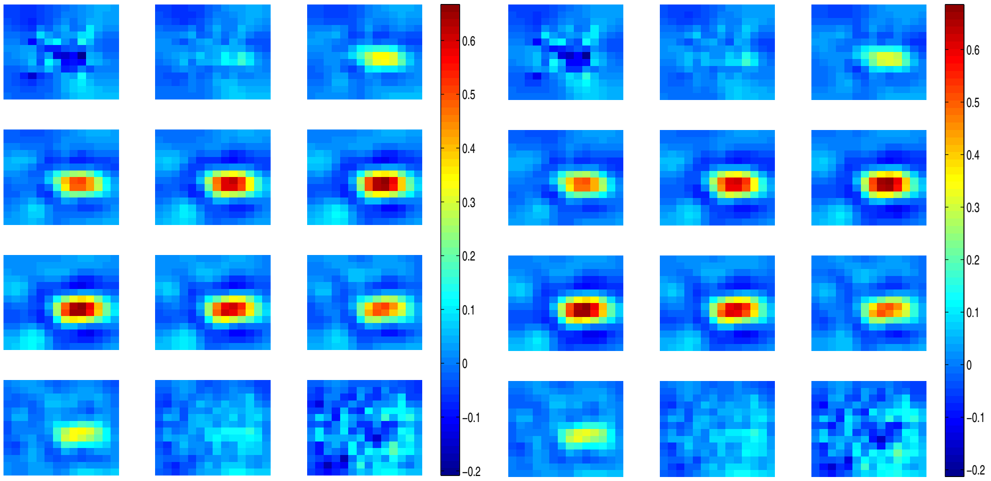
where $E_{\mathbf{f}_b}(\mathbf{f}_a)$ is the error in a reconstruction $\mathbf{f}_a$ with respect to a reconstruction $\mathbf{f}_b$, using the standard 2-norm.

**5.1. Data set 1.** In order to determine how the interpolation error affects the resulting solutions, we first used a simulated data set generated using a known phantom. The data was generated using the image in Figure 5.1(a) as $\mathbf{f}_{real}$ to get $\mathbf{g} = \mathbf{A}\mathbf{f}_{real}$, with random noise added by the Matlab awgn() function at a signal-to-noise ratio of 10dB. Inversions were then run using six different interpolation levels with a number of nodes ranging from 61% $N_{pts}$ down to 3.4% $N_{pts}$. The specific interpolation levels, and number of points computed, are shown in Table 5, along with the relative error of each solution with respect to ground truth.

Images of reconstructions obtained using the (10,10,15) and (15,15,15) interpolation levels, as well as the fully computed matrix, can be seen in Figures 5.1(b)–(d).



(a) Phantom (b) Interpolation Level (10,10,15)



(c) Interpolation Level (15,15,15) (d) Fully Computed

FIG. 5.1. *Results for data set* 1. *A known phantom was used to generate simulated data with* 10*dB white Gaussian noise. Inverse results were obtained for a number of different interpolation levels.* (a) *Phantom used to generate simulated data.* (b) *Result with (*10,10,15*) interpolation level (*578 *explicitly computed points). Absolute error of* 0.7295*, error relative to* (d) *of* 0.1128. (c) *Result with (*15,15,15*) interpolation level (*1098 *computed points). Absolute error of* 0.7406*, error relative to* (d) *of* 0.1035. (d) *Result with fully computed matrix (*11760 *computed points). Absolute error of* 0.7336*. Note that all three constructions are visually almost identical, and that while the relative error between the fully computed and interpolated reconstructions is greater than* 10*, the absolute error changes very little.*

Visual comparison of the three images reveals little if any difference. Analytic comparison results in relative error in the (10,10,15) solution with respect to the fully computed solution of 0.1128, while the (15,15,15) solution exhibits a relative error of 0.1035. Examining the errors with respect to ground truth, however, suggests that despite their differences, the solutions using interpolation are of nearly the same quality as the fully computed solution. Decreasing the number of points computed from

11760 (the full number) to 1098 (the (15,15,15) interpolation level) results in the relative error with respect to ground truth changing only 0.0070 from 0.7336 to 0.7406. Interestingly, the (10,10,15) case, with only 578 points computed, results in a relative error of 0.7295, which is actually a lower error than for the fully computed case. This suggests that, given the ill-posed nature of the problem and the regularization occurring in the inversion, the error induced by the interpolation has little consistent effect upon the absolute error in the solution.

**5.2. Data set 2.** The second data set was generated by ART using an undisclosed forward solver (background $\mu_a$ and $\mu_s$ were provided) and a proprietary noise model. Reconstruction images are shown in Figure 5.2 for the fully computed matrix, as well as (10,10,15) and (15,15,15) interpolation levels.

The results for this data set are similar to those of data set 1. Visually, the three inversion results are nearly identical. Because the ground truth is not known in this case, absolute error values cannot be computed. However, comparing the two interpolated solutions to the fully computed one yields relative differences with respect to the fully computed solution of 0.0947 and 0.0857 for the (15,15,15) and (10,10,15) solutions, respectively. These numbers are similar to those seen in the case of the known phantom, where it was shown that absolute error levels were only slightly perturbed by the use of the approximated $\mathbf{A}_s$.

## 6. Conclusions and future work.

We have presented a method by which the forward matrix $\mathbf{A}$ associated with a certain linearized diffuse optical tomography problem can be efficiently computed and then effectively approximated. Our first step utilizes a change of variables to enable us to represent $\mathbf{A}$ as a core data matrix $\mathbf{A}_s$ and a number of selection matrices $\mathbf{S}_{ij}$. While $\mathbf{A}_s$ is small and dense, the selection matrices are extremely sparse, enabling the entire representation of $\mathbf{A}$ to be stored in significantly less memory. This decomposition also allows for matrix-vector operations upon $\mathbf{A}$ to be performed in a sequential manner, drastically reducing the amount of memory required to perform such operations.

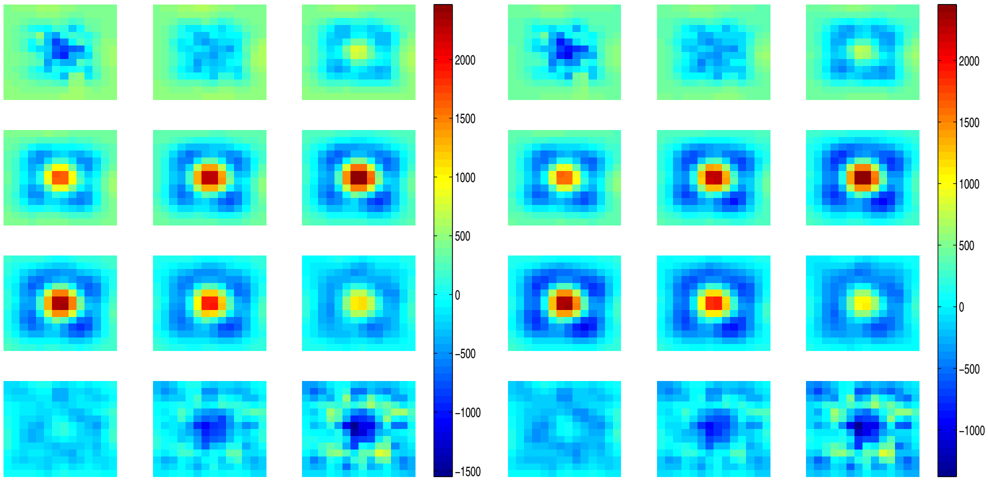
Our simulated results indicate that the use of interpolation to approximate $\mathbf{A}_s$ and thus $\mathbf{A}$ gives accurate solutions. While the solutions using interpolation result in relative errors with respect to the fully computed solution on the order of 0.10, our results indicate that these differences do not significantly affect the error with respect to ground truth.

A further test of this work would be the application of our method to experimentally collected data. Given that the mathematical models presented here are inherently an approximation of reality, there will be an increased mismatch between the data and the model. As such, the model errors introduced by our interpolation scheme should have even less of an effect than they did in the results presented here.

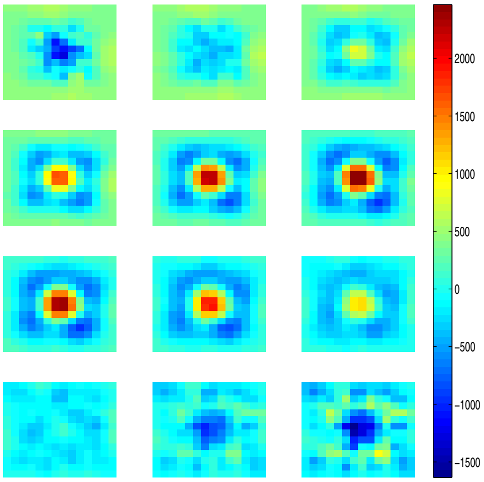
This method also has applications beyond diffuse optical tomography. Any system with a similar invariance to radial angle could potentially be rewritten so as to use our referencing scheme. This includes problems such as continuous wave diffusion imaging, heat transfer in solids, and other problems using omnidirectional sources.

One specific area of application is fluorescence-based optical imaging. Linear models are capable of accurately modeling such systems and have led to systems currently in use for basic in vivo research [21, 20]. Because these systems are optically based using lasers at similar wavelengths, the mathematical details of the diffusion approximation and Green's functions detailed here carry over almost unchanged.

The interpolation could also be applied to other systems, especially those which are linearizations of nonlinear systems. Presuming a reasonably smooth kernel along



(a) Interpolation Level (10,10,15) (b) Interpolation Level (15,15,15)



(c) Fully Computed

Fig. 5.2. *Results for data set* 2*. Data set provided by ART with known background optical parameters, but unknown forward and noise models. The images compare results using different interpolation levels. Note that absolute values for the reconstructions are significantly higher, owing to a lack of information regarding source intensities. Thus the true quantitative values are all multiplied by an unknown scaling factor.* (a) *Result with (*10,10,15*) interpolation level (*578 *explicitly computed points). Error relative to* (c) *of* 0.0857. (b) *Result with (*15,15,15*) interpolation level (*1098 *computed points). Error relative to* (c) *of* 0.0947. (c) *Result with fully computed matrix (*11760 *computed points). Again, all three reconstructions are visually identical and exhibit similar degrees of relative error. The change in absolute error is likely similar to that seen with data set* 1*.*

some dimension, it is feasible that linear approximations of that kernel would result in similarly small changes to the resulting solutions. When computation of individual kernel values is prohibitively expensive, this could lead to significant reductions in required computation. This interpolation could also be investigated to determine its regularizing properties. While our results suggest empirically that any regularizing effect is minimal, a study on the regularizing effects of interpolation-smoothed kernels may be of value.

Finally, further work could seek to extend this type of optimization to the case of systems with structured inhomogeneities. Layered media offer a straightforward extension of our method, while media with more complex structure would require correspondingly more effort. Both would enable our method to be used in a wider range of systems and configurations.

## REFERENCES

[1] S. R. Arridge, *The theoretical basis for the determination of optical pathlengths in tissue*, Phys. Med. Bio., 37 (1992), pp. 1531–1560.

[2] S. R. Arridge, J. C. Hebden, M. Schweiger, F. E. W. Schmidt, M. E. Fry, E. M. C. Hillman, H. Dehghani, and D. T. Delpy, *A method for three-dimensional time-resolved optical tomography*, Int. J. Imaging Systems Technol., 11 (2000), pp. 2–11.

[3] D. A. Boas, D. H. Brooks, E. L. Miller, C. A. DiMarzio, M. Kilmer, R. J. Gaudette, and Q. Zhang, *Imaging the body with diffuse optical tomography*, IEEE Signal Processing Magazine, 18 (2001), pp. 57–75.

[4] D. A. Boas and A. M. Dale, *Simulation study of magnetic resonance imaging-guided cortically constrained diffuse optical tomography of human brain function*, Appl. Optics, 44 (2005), pp. 1957–1968.

[5] S. Chandrasekhar, *Radiative Transfer*, Dover, New York, 1960.

[6] A. Corlu, R. Choe, T. Durduran, K. Lee, M. Schweiger, S. R. Arridge, E. M. C. Hillman, and A. G. Yodh, *Diffuse optical tomography with spectral constraints and wavelength optimization*, Appl. Optics, 44 (2005), pp. 2082–2093.

[7] A. P. Gibson, T. Austion, N. L. Everdell, M. Schweiger, S. R. Arridge, J. G. Meek, J. S. Wyatt, D. T. Delpy, and J. C. Hebden, *Three-dimensional whole-head optical tomography of passive motor evoked responses in the neonate*, Neuroimage, 30 (2006), pp. 521–528.

[8] A. P. Gibson, J. C. Hebden, and S. R. Arridge, *Recent advances in diffuse optical imaging*, Phys. Med. Bio., 50 (2005), pp. R1–R43.

[9] H. L. Graber, Y. Pei, and R. L. Barbour, *Imaging of spatiotemporal coincident states by dc optical tomography*, IEEE Trans. Med. Imag., 21 (2002), pp. 852–866.

[10] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM Monogr. Math. Model. Comput. 4, SIAM, Philadelphia, PA, 1997.

[11] R. C. Haskell, L. O. Svaasand, T.-T. Tsay, T.-C. Feng, M. S. McAdams, and B. J. Tromberg, *Boundary conditions for the diffusion equation in radiative transfer*, J. Opt. Soc. Amer. A, 11 (1994), pp. 2727–2741.

[12] J. C. Hebden and S. R. Arridge, *Imaging through scattering media by the use of an analytical model of perturbation amplitudes in the time domain*, Appl. Optics, 35 (1996), pp. 6788–6796.

[13] M. E. Kilmer, P. C. Hansen, and M. I. Español, *A projection-based approach to general-form Tikhonov regularization*, SIAM J. Sci. Comput., 29 (2007), pp. 315–330.

[14] A. Li, G. Boverman, Y. Zhang, D. Brooks, E. L. Miller, M. E. Kilmer, Q. Zhang, E. M. C. Hillman, and D. A. Boas, *Optimal linear inverse solution with multiple priors in diffuse optical tomography*, Appl. Optics, 44 (2005), pp. 1948–1956.

[15] C. Lindquist, A. Pifferi, R. Berg, S. Andersson-Engels, and S. Svanberg, *Reconstruction of diffuse photon-density wave interference in turbid media from time-resolved transmittance measurements*, Appl. Phys. Lett., 69 (1996), pp. 1674–1676.

[16] V. A. Markel, V. Mital, and J. C. Schotland, *Inverse problem in optical diffusion tomography.* III. *Inversion formulas and singular value decomposition*, J. Opt. Soc. Amer. A, 20 (2003), pp. 890–902.

[17] V. A. Markel and J. C. Schotland, *Scanning paraxial optical tomography*, Optics Lett., 27 (2002), pp. 1123–1125.

[18] V. A. Markel and J. C. Schotland, *Symmetries, inversion formulas, and image reconstruction for optical tomography*, Phys. Rev. E (3), 70 (2004).

[19] V. A. Markel and J. C. Schotland, *Multiple projection optical diffusion tomography with plane wave illumination*, Phys. Med. Bio., 50 (2005), pp. 2351–2364.

[20] V. Ntziachristos, E. A. Schellenberger, J. Ripoll, D. Yessayan, E. Graves, Jr., A. Bogdanov, L. Josephson, and R. Weissleder, *Visualization of antitumor treatment by means of fluorescence molecular tomography with an annexin v-cy*5.5 *conjugate*, Proc.

Natl. Acad. Sci. USA, 101 (2004), pp. 12294–12299.

[21] V. Ntziachristos and R. Weissleder, *Experimental three-dimensional fluorescence reconstruction of diffuse media by use of the normalized Born approximation*, Optics Lett., 26 (2001), pp. 893–895.

[22] M. A. O'Leary, *Imaging with Diffuse Photon Density Waves*, Ph.D. thesis, Physics Department, University of Pennsylvania, Philadelphia, PA, 1996.

[23] C. Paige and M. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[24] M. S. Patterson, B. Chance, and B. C. Wilson, *Time resolved reflectance and transmittance for the noninvasive measurement of tissue optical properties*, Appl. Optics, 28 (1989), p. 2331 ff.

[25] C. H. Schmitz, H. L. Graber, Y. Pei, M. Farber, M. Stewart, R. D. Levina, M. B. Levin, Y. Xu, and R. L. Barbour, *Dynamic studies of small animals with a four-color diffuse optical tomography imager*, Rev. Sci. Inst., 76 (2005).

[26] J. C. Schotland, *Continuous-wave diffusion imaging*, J. Opt. Soc. Amer. A, 14 (1997), pp. 275–279.

[27] M. Schweiger, S. R. Arridge, and I. Nissila, *Gauss–Newton method for image reconstruction in diffuse optical tomography*, Phys. Med. Bio., 50 (2005), pp. 2365–2386.

[28] A. Soubret, J. Ripoll, and V. Ntziachristos, *Accuracy of fluorescent tomography in the presence of heterogeneities: Study of the normalized born ratio*, IEEE Trans. Med. Imag., 24 (2005), pp. 1377–1386.

[29] S. Srinivasan, B. W. Pogue, S. Jiang, H. Dehghani, C. Kogel, W. A. Wells, S. P. Poplack, and K. D. Paulsen, *Near-infrared characterization of breast tumors in vivo using spectrally-constrained reconstruction*, Tech. Cancer Res. Treatment, 4 (2005), pp. 513–526.

[30] Z.-M. Wang, G. Y. Panasyuk, V. A. Markel, and J. C. Schotland, *Experimental demonstration of an analytic method for image reconstruction in optical diffusion tomography with large data sets*, Optics Lett., 30 (2005), pp. 3338–3340.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1083–1100

# ON THE DOUBLING ALGORITHM FOR A (SHIFTED) NONSYMMETRIC ALGEBRAIC RICCATI EQUATION*

CHUN-HUA GUO†, BRUNO IANNAZZO‡, AND BEATRICE MEINI‡

**Abstract.** Nonsymmetric algebraic Riccati equations for which the four coefficient matrices form an irreducible $M$-matrix $M$ are considered. The emphasis is on the case where $M$ is an irreducible singular $M$-matrix, which arises in the study of Markov models. The doubling algorithm is considered for finding the minimal nonnegative solution, the one of practical interest. The algorithm has been recently studied by others for the case where $M$ is a nonsingular $M$-matrix. A shift technique is proposed to transform the original Riccati equation into a new Riccati equation for which the four coefficient matrices form a nonsingular matrix. The convergence of the doubling algorithm is accelerated when it is applied to the shifted Riccati equation.

**Key words.** nonsymmetric algebraic Riccati equation, minimal nonnegative solution, doubling algorithm, convergence acceleration, shift technique

**AMS subject classifications.** 15A24, 15A48, 65F30, 65H10

**DOI.** 10.1137/060660837

## 1. Introduction.

We consider the nonsymmetric algebraic Riccati equation (or NARE)

$$XCX - XD - AX + B = 0, \tag{1.1}$$

where $A, B, C, D$ are real matrices of sizes $m \times m, m \times n, n \times m, n \times n$, respectively, and we assume throughout that

$$M = \begin{bmatrix} D & -C \\ -B & A \end{bmatrix} \tag{1.2}$$

is a nonsingular $M$-matrix or an irreducible singular $M$-matrix. As usual in algebraic Riccati equations theory one associates with (1.1) the matrix

$$H = \begin{bmatrix} D & -C \\ B & -A \end{bmatrix}. \tag{1.3}$$

Some relevant definitions are given below.

For any matrices $A, B \in \mathbb{R}^{m\times n}$, we write $A \geq B (A > B)$ if $a_{ij} \geq b_{ij} (a_{ij} > b_{ij})$ for all $i, j$. A real square matrix $A$ is called a $Z$-matrix if all its off-diagonal elements are nonpositive. Any $Z$-matrix $A$ can be written as $sI - B$ with $B \geq 0$. A $Z$-matrix $A$ is called an $M$-matrix if $s \geq \rho(B)$, where $\rho(\cdot)$ is the spectral radius; it is called a singular $M$-matrix if $s = \rho(B)$ and a nonsingular $M$-matrix if $s > \rho(B)$. Given a square matrix $A$, we will denote by $\sigma(A)$ the set of the eigenvalues of $A$.

---

[*] Received by the editors May 25, 2006; accepted for publication (in revised form) by L. Reichel March 21, 2007; published electronically November 9, 2007.
http://www.siam.org/journals/simax/29-4/66083.html

[†] Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca). This author's research was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

[‡] Dipartimento di Matematica, Università di Pisa, Largo B. Pontecorvo 5, 56127 Pisa, Italy (iannazzo@mail.dm.unipi.it, meini@dm.unipi.it).

The NARE (1.1) has applications in transport theory and Markov models [18, 23, 24]. The solution of practical interest is the minimal nonnegative solution. The equation has attracted much attention recently [2, 4, 6, 8, 9, 11, 13, 14, 15, 19, 20, 22].

Some properties of the NARE (1.1) are summarized below. See [8] and [9] for more details.

THEOREM 1.1. [illegible] (1.1) [illegible] $X$ [illegible] $M$ [illegible] $X > 0$ [illegible] $A - XC$ [illegible] $D - CX$ [illegible] $M$ [illegible] $M$ [illegible] $M$ [illegible] $A - XC$ [illegible] $D - CX$ [illegible] $M$ [illegible]

We will also need the dual equation of (1.1)

$$YBY - YA - DY + C = 0. \tag{1.4}$$

This equation has the same type as (1.1): the matrix

$$\begin{bmatrix} A & -B \\ -C & D \end{bmatrix}$$

is a nonsingular $M$-matrix or an irreducible singular $M$-matrix if and only if the matrix $M$ is so. The minimal nonnegative solution of (1.4) is denoted by $Y$.

A number of numerical methods have been studied for finding the minimal solution $X$. Recently, a doubling algorithm is studied in [15] and is shown to be efficient. The doubling algorithm itself is not new; it was studied in [1], for example. However, the presentation in [15] provides some new information about the algorithm, which makes its analysis easier for the NARE (1.1).

In [15], the discussion is limited to the case where $M$ is a nonsingular $M$-matrix. In the application of the NARE in Markov chains, however, the most important case is the one where $M$ is an irreducible singular $M$-matrix with zero row sums. So in this paper, we will assume that $M$ is an irreducible (singular or nonsingular) $M$-matrix, with the emphasis on the singular case.

We show the applicability and convergence properties of the structure preserving doubling algorithm of [15] when $M$ is singular. In particular, we show that the algorithm has quadratic convergence when 0 is a simple eigenvalue of $H$. From the numerical experiments performed so far, the doubling algorithm shows a linear convergence of rate 1/2 if 0 has algebraic multiplicity equal to 2.

We introduce an alternative approach to treat the singular case based on a shift technique. The shift consists in performing a rank-one correction of the matrix $H$ which moves one zero eigenvalue to a suitable nonzero real number. We construct a new Riccati equation associated with the shifted $H$, which has the same solution $X$ of the original one, while the coefficients of the new Riccati equation form a nonsingular matrix if 0 is a simple eigenvalue of $H$.

We analyze the structure preserving doubling algorithm for the new Riccati equation and show that its convergence is faster (when no breakdown is encountered) than the convergence of the same algorithm applied to the original equation. In particular, when 0 is a double eigenvalue of $H$, the doubling algorithm applied to the new equation is shown to have quadratic convergence.

Numerical results show the effectiveness of the shift technique.

The paper is organized as follows. In section 2 we recall some properties of the Riccati equations and nonnegative matrices. In sections 3 and 4 we show that the structure preserving doubling algorithm of [15] can also be applied to the case where $M$ is irreducible singular, and we show convergence results. In section 5 we present the shift technique. In section 6 we analyze the doubling algorithm applied to the new Riccati equation. In section 7 we show some numerical experiments.

**2. Preliminaries.** When $M$ is an irreducible singular $M$-matrix, by the Perron–Frobenius theory 0 is a simple eigenvalue and there are positive vectors $u$ and $v$ such that

$$u^TM = 0, \quad Mv = 0, \tag{2.1}$$

and the vectors $u$ and $v$ are each unique up to a scalar multiple.

For any solution $S$ of the Riccati equation (1.1), the matrix $H$ of (1.3) satisfies

$$H\begin{bmatrix} I \\ S \end{bmatrix} = \begin{bmatrix} I \\ S \end{bmatrix} R,$$

where $R = D - CS$. The eigenvalues of the matrix $R$ are a subset of the eigenvalues of $H$.

Since $H = JM$, where $J = \left[\begin{smallmatrix} I_n & 0 \\ 0 & -I_m \end{smallmatrix}\right]$, then $H$ has a one-dimensional kernel, and $u^TJ$ and $v$ are the left and right eigenvectors corresponding to the eigenvalue 0.

Writing $u^T = \left(u_1^T, u_2^T\right)$ and $v^T = \left(v_1^T, v_2^T\right)$, with $u_1, v_1 \in \mathbb{R}^n$ and $u_2, v_2 \in \mathbb{R}^m$, one can define $\mu = u_1^Tv_1 - u_2^Tv_2$.

The number $\mu$ determines some properties of the equation. Depending on the sign of $\mu$ and following a Markov chain terminology, one can classify the Riccati equations associated with an irreducible singular $M$-matrix in three categories: a Riccati equation will be called

(a) [illegible] if $\mu > 0$;

(b) [illegible] if $\mu = 0$;

(c) [illegible] if $\mu < 0$.

The [illegible] case, i.e., the case $\mu \approx 0$, deserves particular attention, since it corresponds to an ill-conditioned zero eigenvalue for the matrix $H$. In fact if $u$ and $v$ are normalized such that $\|u\|_2 = \|v\|_2 = 1$, then $1/|\mu|$ is the condition number of the zero eigenvalue for the matrix $H$ (see [7]).

In fluid queues problems, $v$ coincides with the vector of ones, which will be denoted by $e$. In general, $v$ and $u$ can be computed by performing an LU factorization of the matrix $M$ and solving two triangular linear systems.

The next results concern $\mu$ and are proved or follow easily from results shown in [8, 9, 12].

THEOREM 2.1. [illegible] $M$ [illegible] $M$ [illegible] $X$ [illegible] $Y$ [illegible] (1.1) [illegible] (1.4) [illegible]

(a) [illegible] $\mu > 0$ [illegible] $Xv_1 = v_2$ [illegible] $Yv_2 < v_1$.

(b) [illegible] $\mu = 0$ [illegible] $Xv_1 = v_2$ [illegible] $Yv_2 = v_1$.

(c) [illegible] $\mu < 0$ [illegible] $Xv_1 < v_2$ [illegible] $Yv_2 = v_1$

THEOREM 2.2. [illegible] $M$ [illegible] $M$ [illegible] $\lambda_1, \dots, \lambda_{m+n}$ [illegible] $H = \operatorname{diag}(I_n, -I_m)M$ [illegible] $\lambda_n$ [illegible] $\lambda_{n+1}$ [illegible]

$$\mathrm{Re}\lambda_{n+m} \le \cdots \le \mathrm{Re}\lambda_{n+2} < \lambda_{n+1} \le 0 \le \lambda_n < \mathrm{Re}\lambda_{n-1} \le \cdots \le \mathrm{Re}\lambda_1.$$

[illegible] $X$ [illegible] (1.1) [illegible] $Y$ [illegible] (1.4) [illegible] $\sigma(D - CX) = \{\lambda_1, \dots, \lambda_n\}$ [illegible] $\sigma(A - XC) = \sigma(A - BY) = \{-\lambda_{n+1}, \dots, -\lambda_{n+m}\}$

[illegible] $M$ [illegible] $\mu > 0$ [illegible] $\lambda_n = 0, \lambda_{n+1} < 0$. [illegible] $\mu = 0$ [illegible] $\lambda_n = \lambda_{n+1} = 0$ [illegible] $\mu < 0$ [illegible] $\lambda_n > 0, \lambda_{n+1} = 0$

In what follows, we will need some basic results about $M$-matrices. The first result can be found in [3], for example.

THEOREM 2.3. [illegible] $Z$ [illegible] $A$ [illegible]

(a) $A$ [illegible] $M$ [illegible]

(b) $A^{-1} \geq 0$

(c) $Av > 0$ [illegible] $v > 0$

(d) [illegible] $A$ [illegible]

The equivalence of (a) and (c) in Theorem 2.3 implies the next result.

LEMMA 2.4. [illegible] $A$ [illegible] $M$ [illegible] $B \geq A$ [illegible] $Z$ [illegible] $B$ [illegible] $M$ [illegible]

Most of the statements in the following result are also well known.

LEMMA 2.5. [illegible] $M$ [illegible] $M$ [illegible] $M$ [illegible] $M$ [illegible]

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

[illegible] $M_{11}$ [illegible] $M_{22}$ [illegible] $M_{11}$ [illegible] $M_{22}$ [illegible] $M$ [illegible] $M_{11}$ ( [illegible] $M_{22}$) [illegible] $M$ [illegible] $M$ [illegible] ( [illegible] $M$) [illegible] $M$ [illegible]

[illegible] 2.6. The last statement in Lemma 2.5 follows from Theorem 2.3 of [21], where the irreducibility of the Schur complement is proved for any irreducible singular $M$-matrix of the form $I - P$ with $P$ stochastic. For a general irreducible $M$-matrix $M$, we have $M = s(I - B)$ for some scalar $s > 0$ and some irreducible $B \geq 0$ with $\rho(B) \leq 1$. Note that if we replace $B$ with a stochastic matrix with the same nonzero pattern, there will be no change of the nonzero pattern in the Schur complement. In other words, the irreducibility will not change.

## 3. The doubling algorithm.

In this section we review the structure preserving doubling algorithm (SDA) for the NARE (1.1) and show that the algorithm is well defined when $M$ is an irreducible singular $M$-matrix. When $M$ is a nonsingular $M$-matrix, the algorithm has already been shown to be well defined in [15], although the selection of a parameter in the algorithm is slightly more restrictive in [15].

For the minimal nonnegative solution $X$ of the NARE (1.1), we have

$$H \begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix} R, \tag{3.1}$$

where $H$ is defined in (1.3) and $R = D - CX$.

Using the Cayley transform

$$\mathcal{C}_\gamma : z \to \frac{z-\gamma}{z+\gamma}, \tag{3.2}$$

where $\gamma > 0$ is a positive scalar, we can transform (3.1) into

$$(H - \gamma I) \begin{bmatrix} I \\ X \end{bmatrix} = (H + \gamma I) \begin{bmatrix} I \\ X \end{bmatrix} R_\gamma, \tag{3.3}$$

where

$$R_\gamma = \mathcal{C}_\gamma(R) = (R + \gamma I)^{-1}(R - \gamma I).$$

Note that $R + \gamma I$ is nonsingular since $R$ is an $M$-matrix by Theorem 1.1. For any $\gamma > 0$, the matrix $M_\gamma = M + \gamma I$ is a nonsingular $M$-matrix. So

$$A_\gamma = A + \gamma I, \quad D_\gamma = D + \gamma I$$

are nonsingular $M$-matrices. Let

$$W_\gamma = A_\gamma - BD_\gamma^{-1}C, \quad V_\gamma = D_\gamma - CA_\gamma^{-1}B \tag{3.4}$$

be the Schur complements of $D_\gamma$ and $A_\gamma$, respectively, in $M_\gamma$. They are both nonsingular $M$-matrices by Lemma 2.5. It is shown in [15] that (3.3) can be reduced to

$$K \begin{bmatrix} I \\ X \end{bmatrix} = L \begin{bmatrix} I \\ X \end{bmatrix} R_\gamma, \tag{3.5}$$

by premultiplying both sides of (3.3) with a proper nonsingular matrix, where

$$K = \begin{bmatrix} E_\gamma & 0 \\ -H_\gamma & I \end{bmatrix}, \quad L = \begin{bmatrix} I & -G_\gamma \\ 0 & F_\gamma \end{bmatrix},$$

with

$$\begin{aligned} E_\gamma &= I - 2\gamma V_\gamma^{-1}, \quad F_\gamma = I - 2\gamma W_\gamma^{-1}, \\ G_\gamma &= 2\gamma D_\gamma^{-1} C W_\gamma^{-1}, \quad H_\gamma = 2\gamma W_\gamma^{-1} B D_\gamma^{-1}. \end{aligned} \tag{3.6}$$

Similarly, for the minimal nonnegative solution $Y$ of the NARE (1.4), we have

$$(H - \gamma I) \begin{bmatrix} Y \\ I \end{bmatrix} S_\gamma = (H + \gamma I) \begin{bmatrix} Y \\ I \end{bmatrix} \tag{3.7}$$

and then

$$K \begin{bmatrix} Y \\ I \end{bmatrix} S_\gamma = L \begin{bmatrix} Y \\ I \end{bmatrix}, \tag{3.8}$$

where $S_\gamma = (S + \gamma I)^{-1}(S - \gamma I)$ with $S = A - BY$ being an $M$-matrix.

The doubling algorithm presented in [15] is the following, where the sequences $\{H_k\}$ and $\{G_k\}$ are going to approximate $X$ and $Y$, respectively.

ALGORITHM 3.1.

$$\begin{aligned} &E_0 = E_\gamma, \quad F_0 = F_\gamma, \quad G_0 = G_\gamma, \quad H_0 = H_\gamma, \\ &E_{k+1} = E_k (I - G_k H_k)^{-1} E_k, \\ &F_{k+1} = F_k (I - H_k G_k)^{-1} F_k, \\ &G_{k+1} = G_k + E_k (I - G_k H_k)^{-1} G_k F_k, \\ &H_{k+1} = H_k + F_k (I - H_k G_k)^{-1} H_k E_k. \end{aligned}$$

In this section we show that the algorithm is well defined. The convergence behavior of the algorithm will be studied in the next section.

THEOREM 3.2. [illegible] $M$ [illegible] $M$ [illegible]

$$\gamma \geq \max\left\{ \max_{1 \leq i \leq m} a_{ii}, \max_{1 \leq i \leq n} d_{ii} \right\},$$

[illegible] $a_{ii}$ [illegible] $d_{ii}$ [illegible] $A$ [illegible] $D$ [illegible] $E_\gamma$ $F_\gamma$ $R_\gamma$ $S_\gamma < 0$ [illegible] $0 \le G_\gamma < Y, 0 \le H_\gamma < X, G_\gamma, H_\gamma \neq 0$ $I - G_\gamma H_\gamma$ [illegible] $I - H_\gamma G_\gamma$ [illegible] $M$ [illegible]

[illegible] We have

$$E_\gamma = I - 2\gamma V_\gamma^{-1} = V_\gamma^{-1}(V_\gamma - 2\gamma I).$$

Since $V_\gamma$ is the Schur complement of $A_\gamma$ in the irreducible nonsingular $M$-matrix $M_\gamma$, it is also an irreducible nonsingular $M$-matrix by Lemma 2.5. So $V_\gamma^{-1} > 0$ (see [3]). Since $\gamma \ge \max_{1\le i\le n} d_{ii}$, then $V_\gamma - 2\gamma I = -\gamma I + D - CA_\gamma^{-1}B \le 0$. Since $V_\gamma - 2\gamma I$ is irreducible, it has no zero columns, whence $E_\gamma < 0$.

Since $R = D - CX$ is an irreducible $M$-matrix by Theorem 1.1, $R + \gamma I$ is an irreducible nonsingular $M$-matrix and $(R + \gamma I)^{-1} > 0$, for any $\gamma > 0$. For $\gamma \ge \max_{1\le i\le n} d_{ii}$, one has $R - \gamma I = D - \gamma I - CX \le 0$. Since $R - \gamma I$ is irreducible and thus has no zero columns, then it follows that $R_\gamma = (R + \gamma I)^{-1}(R - \gamma I) < 0$.

Similarly, using $\gamma \ge \max_{1\le i\le m} a_{ii}$, we can prove that $F_\gamma < 0$, $S_\gamma < 0$.

It is clear that $G_\gamma, H_\gamma \ge 0$. Since $M$ is irreducible, then $B, C \neq 0$, whence it follows that $H_\gamma, G_\gamma \neq 0$. It is shown in [15] that $X - H_\gamma = F_\gamma X R_\gamma$. Since $F_\gamma X R_\gamma > 0$, we have $0 \le H_\gamma < X$. Similarly, we have $0 \le G_\gamma < Y$, so that $0 \le G_\gamma H_\gamma < YX$. By Theorem 2.1 we have $YXv_1 \le v_1$. Thus $\rho(G_\gamma H_\gamma) < \rho(YX) \le 1$ by the Perron–Frobenius theory. Therefore, $I - G_\gamma H_\gamma$ is a nonsingular $M$-matrix by Theorem 2.3. Similarly, $I - H_\gamma G_\gamma$ is a nonsingular $M$-matrix. ▯

THEOREM 3.3. [illegible] $M$ [illegible] $M$ [illegible] $k \ge 1$ $E_k, F_k > 0$ $H_{k-1} < H_k < X$ $G_{k-1} < G_k < Y$ [illegible] $I - H_k G_k, I - G_k H_k$ [illegible] $M$ [illegible]

[illegible] For any nonnegative matrices $U, V, W$ such that $UVW$ is defined, if $U, W > 0$ and $V \neq 0$, then $UVW > 0$. Since $E_0, F_0 < 0$ and $I - G_0H_0, I - H_0G_0$ are nonsingular $M$-matrices, we have

$$E_1, F_1 > 0, \quad H_1 > H_0, \quad G_1 > G_0.$$

For the doubling algorithm, it is shown in [15] that

$$X - H_1 = F_1 X R_\gamma^2, \quad Y - G_1 = E_1 Y S_\gamma^2.$$

Thus, $H_1 < X$ and $G_1 < Y$. Then, as in the proof of Theorem 3.2, $I - G_1H_1$ and $I - H_1G_1$ are nonsingular $M$-matrices. The statements in the theorem are now easily proved by induction. ▯

## 4. Convergence of the doubling algorithm.

For the doubling algorithm, we have (see [15])

$$X - H_k = F_k X R_\gamma^{2^k}, \quad Y - G_k = E_k Y S_\gamma^{2^k},$$

$$E_k = (I - G_kX)R_\gamma^{2^k} \le R_\gamma^{2^k}, \quad F_k = (I - H_kY)S_\gamma^{2^k} \le S_\gamma^{2^k} \tag{4.1}$$

for each $k \ge 1$. So

$$X - H_k = (I - H_kY)S_\gamma^{2^k} X R_\gamma^{2^k} \le S_\gamma^{2^k} X R_\gamma^{2^k}, \tag{4.2}$$

$$Y - G_k = (I - G_kX)R_\gamma^{2^k} Y S_\gamma^{2^k} \le R_\gamma^{2^k} Y S_\gamma^{2^k}. \tag{4.3}$$

When $M$ is an irreducible nonsingular $M$-matrix, we have $\rho(R_\gamma) < 1$ and $\rho(S_\gamma) < 1$. It follows that $\{H_k\}$ converges to $X$, $\{G_k\}$ converges to $Y$, $\{E_k\}$ and $\{F_k\}$ converge to 0, all quadratically. This result is shown in [15] under the assumption that $\gamma > \max\{\max a_{ii}, \max d_{ii}\}$, but without the irreducibility assumption. Here we would like to allow $\gamma = \max\{\max a_{ii}, \max d_{ii}\}$, since this $\gamma$ will be shown to be optimal in some sense. From (4.2) and (4.3), we also have

$$\limsup_{k\to\infty} \sqrt[2^k]{\|H_k - X\|} \le \rho(R_\gamma)\rho(S_\gamma), \tag{4.4}$$

$$\limsup_{k\to\infty} \sqrt[2^k]{\|G_k - Y\|} \le \rho(R_\gamma)\rho(S_\gamma). \tag{4.5}$$

THEOREM 4.1. [illegible] $M$ [illegible] $M$ [illegible]
(a) [illegible] $\mu > 0$ [illegible] $\{H_k\}$ ($\{G_k\}$) [illegible] $X$ ($Y$) [illegible]

$$\limsup_{k\to\infty} \sqrt[2^k]{\|H_k - X\|} \le \rho(S_\gamma) < 1, \quad \limsup_{k\to\infty} \sqrt[2^k]{\|G_k - Y\|} \le \rho(S_\gamma),$$

$\{F_k\}$ [illegible] 0 [illegible]

$$\limsup_{k\to\infty} \sqrt[2^k]{\|F_k\|} \le \rho(S_\gamma),$$

[illegible] $\{E_k\}$ [illegible]
(b) [illegible] $\mu < 0$ [illegible] $\{H_k\}$ ($\{G_k\}$) [illegible] $X$ ($Y$) [illegible]

$$\limsup_{k\to\infty} \sqrt[2^k]{\|H_k - X\|} \le \rho(R_\gamma) < 1, \quad \limsup_{k\to\infty} \sqrt[2^k]{\|G_k - Y\|} \le \rho(R_\gamma),$$

$\{E_k\}$ [illegible] 0 [illegible]

$$\limsup_{k\to\infty} \sqrt[2^k]{\|E_k\|} \le \rho(R_\gamma),$$

[illegible] $\{F_k\}$ [illegible]
(c) [illegible] $\mu = 0$ [illegible] $\{H_k\}$ [illegible] $X$ $\{G_k\}$ [illegible] $Y$ [illegible] $\{E_k\}, \{F_k\}$ [illegible]

[illegible] When $\mu > 0$, one has $\rho(S_\gamma) < 1$ and $\rho(R_\gamma) = 1$. Moreover, $-1$ is a simple eigenvalue of $R_\gamma$ and there are no other eigenvalues on the unit circle. When $\mu < 0$, one has $\rho(R_\gamma) < 1$ and $\rho(S_\gamma) = 1$. Moreover, $-1$ is a simple eigenvalue of $S_\gamma$ and there are no other eigenvalues on the unit circle. The statements in (a) and (b) are then valid in view of (4.1), (4.2), and (4.3).

When $\mu = 0$, one has $\rho(R_\gamma) = 1$. Moreover, $-1$ is a simple eigenvalue of $R_\gamma$ and there are no other eigenvalues on the unit circle. Also, $\rho(S_\gamma) = 1$, $-1$ is a simple eigenvalue of $S_\gamma$ and there are no other eigenvalues on the unit circle. The boundedness of $\{E_k\}$ and $\{F_k\}$ then follows immediately. However, from (4.2) and (4.3), we cannot see the convergence of $\{H_k\}$ and $\{G_k\}$ to $X$ and $Y$, respectively. So we will take a different approach.

For the minimal solution $X$ of (1.1), we have from (3.5)

$$E_\gamma = (I - G_\gamma X)R_\gamma, \quad X - H_\gamma = F_\gamma X R_\gamma. \tag{4.6}$$

Since $0 \le G_\gamma X < YX$, $I - G_\gamma X$ is a nonsingular $M$-matrix as in the proof of Theorem 3.2. Eliminating $R_\gamma$ in (4.6) gives

$$X = F_\gamma X (I - G_\gamma X)^{-1} E_\gamma + H_\gamma. \tag{4.7}$$

We now consider the basic fixed-point iteration for (4.7):

$$X_{k+1} = F_\gamma X_k (I - G_\gamma X_k)^{-1} E_\gamma + H_\gamma, \quad X_0 = 0. \tag{4.8}$$

It is easily proved by induction that $I - G_\gamma X_k$ is a nonsingular $M$-matrix and $X_k \le X_{k+1} \le X$ for all $k \ge 0$. Therefore, $\lim X_k = \widehat{X}$ with $0 \le \widehat{X} \le X$. Since $I - G_\gamma X$ is a nonsingular $M$-matrix, so is $I - G_\gamma \widehat{X}$. Thus we have

$$\widehat{X} = F_\gamma \widehat{X} (I - G_\gamma \widehat{X})^{-1} E_\gamma + H_\gamma. \tag{4.9}$$

We are going to show that $\widehat{X} = X$. Let $\widehat{R}_\gamma = (I - G_\gamma \widehat{X})^{-1} E_\gamma$, then

$$E_\gamma = (I - G_\gamma \widehat{X}) \widehat{R}_\gamma, \quad \widehat{X} - H_\gamma = F_\gamma \widehat{X} \widehat{R}_\gamma.$$

So instead of (3.5) we have

$$K \begin{bmatrix} I \\ \widehat{X} \end{bmatrix} = L \begin{bmatrix} I \\ \widehat{X} \end{bmatrix} \widehat{R}_\gamma, \tag{4.10}$$

which can be transformed back to

$$(H - \gamma I) \begin{bmatrix} I \\ \widehat{X} \end{bmatrix} = (H + \gamma I) \begin{bmatrix} I \\ \widehat{X} \end{bmatrix} \widehat{R}_\gamma. \tag{4.11}$$

If $G_\gamma \widehat{X} = G_\gamma X$, then $\widehat{R}_\gamma = R_\gamma$ and $I - \widehat{R}_\gamma$ is nonsingular. If $G_\gamma \widehat{X} \ne G_\gamma X$, then we have $0 < (I - G_\gamma \widehat{X})^{-1}(-E_\gamma) \le (I - G_\gamma X)^{-1}(-E_\gamma)$ and $(I - G_\gamma \widehat{X})^{-1}(-E_\gamma) \ne (I - G_\gamma X)^{-1}(-E_\gamma)$ since $E_\gamma < 0$. It follows from the Perron–Frobenius theory that $\rho((I - G_\gamma \widehat{X})^{-1}(-E_\gamma)) < \rho((I - G_\gamma X)^{-1}(-E_\gamma))$. Thus $\rho(\widehat{R}_\gamma) < \rho(R_\gamma) = 1$, and again $I - \widehat{R}_\gamma$ is nonsingular. Now (4.11) can be rewritten as

$$H \begin{bmatrix} I \\ \widehat{X} \end{bmatrix} = \begin{bmatrix} I \\ \widehat{X} \end{bmatrix} \gamma (I + \widehat{R}_\gamma)(I - \widehat{R}_\gamma)^{-1}. \tag{4.12}$$

Thus $\widehat{X}$ is also a nonnegative solution of (1.1). Since $X$ is minimal we have $\widehat{X} = X$ and so $\lim X_k = X$. Now, the sequence $\{H_k\}$ produced by the doubling algorithm is such that $H_k = X_{2^k}$ (see [1]). Therefore, $\lim H_k = X$, as required. The proof of $\lim G_k = Y$ is similar. □

4.2. When $\mu = 0$, the convergence of the doubling algorithm is not quadratic in general since the convergence of $\{X_k\}$ in (4.8) is sublinear in general. But the relation $H_k = X_{2^k}$ itself says that the convergence of the doubling algorithm is much faster than the basic fixed point iteration. The convergence in this case has been observed to be linear with rate 1/2.

4.3. We thank one referee for pointing out that the convergence of the doubling algorithm for the case $\mu = 0$ has recently been proved in [5] to be at least linear with rate 1/2.

In the doubling algorithm, we have the freedom to choose the parameter $\gamma$. In view of (4.4) and (4.5), the next result says that $\gamma = \max\{\max a_{ii}, \max d_{ii}\}$ is optimal, in some sense, for the doubling algorithm.

THEOREM 4.4. $\gamma \ge \max\{\max a_{ii}, \max d_{ii}\}$ $\rho(R_\gamma)$ $\rho(S_\gamma)$ $\gamma$

[illegible] Since $R = D - CX$ is an irreducible $M$-matrix, it can be written in the form $sI - N$, where $N \geq 0$ is irreducible. It follows from the Perron–Frobenius theorem that there is a positive vector $v$ such that $Rv = \lambda_n v$. Now

$$-R_\gamma v = (\gamma I + R)^{-1}(\gamma I - R)v = (\gamma + \lambda_n)^{-1}(\gamma - \lambda_n)v.$$

Since $-R_\gamma > 0$, it follows from the Perron–Frobenius theory that $\rho(R_\gamma) = \rho(-R_\gamma) = (\gamma + \lambda_n)^{-1}(\gamma - \lambda_n)$, which is a nondecreasing function of $\gamma$. Similarly, $\rho(S_\gamma)$ is a nondecreasing function of $\gamma$. ☐

**5. A shift technique.** In this section we assume that $M$ is an irreducible singular $M$-matrix. The vectors $u$ and $v$ are as in (2.1), and we have three cases: $\mu > 0$, $\mu = 0$, and $\mu < 0$. The next result shows that the case $\mu < 0$ is easily reduced to the case $\mu > 0$.

LEMMA 5.1. [illegible] $X$ [illegible] (1.1) [illegible] $Z = X^T$ [illegible]

$$ZC^TZ - ZA^T - D^TZ + B^T = 0. \tag{5.1}$$

[illegible] (1.1) [illegible] (5.1) [illegible]

[illegible] The first statement is easily shown by transposing on both sides of the equation. The $M$-matrix corresponding to (5.1) is

$$M_t = \begin{bmatrix} A^T & -C^T \\ -B^T & D^T \end{bmatrix}.$$

Since

$$\begin{bmatrix} v_2^T & v_1^T \end{bmatrix} M_t = 0, \quad M_t \begin{bmatrix} u_2 \\ u_1 \end{bmatrix} = 0,$$

the second statement follows readily. ☐

[illegible] 5.2. When $\mu \leq 0$, from the above proof and Theorem 2.1 we know that the minimal nonnegative solution $X$ of (1.1) is such that $X^Tu_2 = u_1$, or in other words, $u_2^TX = u_1^T$.

From now on, we assume that $\mu \geq 0$.

Our shift technique will be based on the following result (see also [17]).

LEMMA 5.3. [illegible] $T$ [illegible] $n \times n$ [illegible] $Tw = 0$ [illegible] $w$ [illegible] $r$ [illegible] $r^Tw = 1$ [illegible] $\eta$ [illegible]

$$\widehat{T} = T + \eta w r^T$$

[illegible] $T$ [illegible] $\widehat{T}$ [illegible] $\eta$

[illegible] We may easily verify that $(\widehat{T} - \lambda I)\lambda I = (T - \lambda I)(\lambda I - \eta w r^T)$ for any complex number $\lambda$. Taking determinants, one has that

$$\lambda^n \det(\widehat{T} - \lambda I) = \lambda^{n-1}(\lambda - \eta)\det(T - \lambda I),$$

from which the proof follows. ☐

We now construct a rank-one modification of the matrix $H$ in (1.3):

$$\widehat{H} = H + \eta v p^T, \tag{5.2}$$

where $\eta > 0$ is a scalar and $p \geq 0$ is a vector with $p^T v = 1$. Since $H$ is a singular matrix with $Hv = 0$, we know from Lemma 5.3 that the eigenvalues of $\widehat{H}$ are those of $H$ except that one zero eigenvalue of $H$ is replaced by $\eta$.

We write $p^T = \left(p_1^T, p_2^T\right)$ and

$$\widehat{H} = \begin{bmatrix} \widehat{D} & -\widehat{C} \\ \widehat{B} & -\widehat{A} \end{bmatrix}, \quad \widehat{M} = \begin{bmatrix} \widehat{D} & -\widehat{C} \\ -\widehat{B} & \widehat{A} \end{bmatrix},$$

where

$$\widehat{D} = D + \eta v_1 p_1^T, \quad \widehat{C} = C - \eta v_1 p_2^T,$$

$$\widehat{B} = B + \eta v_2 p_1^T, \quad \widehat{A} = A - \eta v_2 p_2^T.$$

Corresponding to $\widehat{M}$ we define the new NARE

$$Z\widehat{C}Z - Z\widehat{D} - \widehat{A}Z + \widehat{B} = 0. \tag{5.3}$$

We have the following important property about the NARE (5.3).

THEOREM 5.4. [illegible] $\mu \geq 0$ [illegible] $Z = X$ [illegible] (5.3) [illegible] $\sigma(\widehat{D} - \widehat{C}X) = \{\lambda_1, \ldots, \lambda_{n-1}, \eta\}$ [illegible] $X$ [illegible] (1.1)

[illegible] Observe that

$$X\widehat{C}X - X\widehat{D} - \widehat{A}X + \widehat{B} = XCX - XD - AX + B - \eta(Xv_1 - v_2)(p_2^T X + p_1^T).$$

Since $X$ is a solution of (1.1) and $Xv_1 = v_2$ by Theorem 2.1, $X$ is also a solution of the shifted equation (5.3). We have $\widehat{D} - \widehat{C}X = D - CX + \eta v_1(p_1^T + p_2^T X)$. Since $(D - CX)v_1 = Dv_1 - Cv_2 = 0$ and $(p_1^T + p_2^T X)v_1 = p^T v = 1$, the eigenvalues of $\widehat{D} - \widehat{C}X$ are $\lambda_1, \ldots, \lambda_{n-1}, \eta$ by Theorem 2.2 and Lemma 5.3. ☐

In what follows we will show that the dual equation of (5.3) has a solution $\widehat{Y}$ such that the eigenvalues of $-(\widehat{A} - \widehat{B}\widehat{Y})$ are the remaining eigenvalues of $\widehat{H}$: $\lambda_{n+1}, \ldots, \lambda_{n+m}$.

LEMMA 5.5. [illegible]

$$W = \begin{bmatrix} \eta p_1^T(v_1 - Yv_2) & \eta(p_1^T Y + p_2^T) \\ (B + \eta v_2 p_1^T)(v_1 - Yv_2) & -(A - BY - \eta v_2(p_1^T Y + p_2^T)) \end{bmatrix}$$

[illegible] $\eta, \lambda_{n+1}, \ldots, \lambda_{n+m}$

[illegible] We have

$$W = W_0 + \eta \begin{bmatrix} 1 \\ v_2 \end{bmatrix} \begin{bmatrix} p_1^T(v_1 - Yv_2) & p_1^T Y + p_2^T \end{bmatrix},$$

where

$$W_0 = \begin{bmatrix} 0 & 0 \\ B(v_1 - Yv_2) & -(A - BY) \end{bmatrix}.$$

The eigenvalues of $W_0$ are $0, \lambda_{n+1}, \ldots, \lambda_{n+m}$ by Theorem 2.2. Since

$$W_0 \begin{bmatrix} 1 \\ v_2 \end{bmatrix} = 0, \quad \begin{bmatrix} p_1^T(v_1 - Yv_2) & p_1^T Y + p_2^T \end{bmatrix} \begin{bmatrix} 1 \\ v_2 \end{bmatrix} = p^T v = 1,$$

the eigenvalues of $W$ are $\eta, \lambda_{n+1}, \ldots, \lambda_{n+m}$ by Lemma 5.3. ☐

LEMMA 5.6. [illegible] $\mu > 0$ [illegible] $f$ [illegible] $(1, f^T)$ [illegible] $W$ [illegible] $\eta$

[illegible] When $\mu > 0$, we have $Yv_2 < v_1$ by Theorem 2.1. Since $A - BY$ is an irreducible $M$-matrix, $A - BY - \eta v_2(p_1^T Y + p_2^T)$ is an irreducible $Z$-matrix. Since $\eta(p_1^T Y + p_2^T) \neq 0$ and $(B + \eta v_2 p_1^T)(v_1 - Yv_2) \neq 0$, the matrix $W$ is irreducible by a simple graph argument. It is clear that $W$ can be written in the form $N - sI$, where $N \geq 0$ is irreducible. The result then follows from the Perron–Frobenius theorem. ☐

LEMMA 5.7. [illegible] $\mu > 0$ [illegible] $\widehat{Y} = Y + (Yv_2 - v_1)f^T$ [illegible] (5.3)

[illegible] Let $\mathcal{R}(Z) = ZBZ - ZA - DZ + C$ and $\widehat{\mathcal{R}}(Z) = Z\widehat{B}Z - Z\widehat{A} - \widehat{D}Z + \widehat{C}$. We are to show $\widehat{\mathcal{R}}(\widehat{Y}) = 0$. Since $\mathcal{R}(Y) = 0$, we have

$$\widehat{\mathcal{R}}(\widehat{Y}) = (\widehat{\mathcal{R}}(\widehat{Y}) - \mathcal{R}(\widehat{Y})) + (\mathcal{R}(\widehat{Y}) - \mathcal{R}(Y)).$$

A straightforward computation shows that

$$\begin{aligned}\widehat{\mathcal{R}}(\widehat{Y}) - \mathcal{R}(\widehat{Y}) &= \eta(\widehat{Y}v_2 - v_1)(p_1^T \widehat{Y} + p_2^T) \\ &= \eta(Yv_2 - v_1)(1 + f^T v_2)(p_1^T Y + p_2^T + p_1^T(Yv_2 - v_1)f^T).\end{aligned}$$

Also, we have

$$\mathcal{R}(\widehat{Y}) - \mathcal{R}(Y) = (Yv_2 - v_1)(f^T B(Yv_2 - v_1)f^T - f^T(A - BY)),$$

where we have used the fact that

$$\begin{aligned}(D - YB)(Yv_2 - v_1) &= -Dv_1 + YBv_1 + (DY - YBY)v_2 \\ &= -Dv_1 + YBv_1 + (C - YA)v_2 = 0.\end{aligned}$$

Thus, to show $\widehat{\mathcal{R}}(\widehat{Y}) = 0$, we need only check

$$\begin{aligned}&f^T(B + \eta v_2 p_1^T)(Yv_2 - v_1)f^T + \eta p_1^T(Yv_2 - v_1)f^T \\ &\quad -f^T(A - BY - \eta v_2(p_1^T Y + p_2^T)) + \eta(p_1^T Y + p_2^T) = 0,\end{aligned}$$

which is true by the choice of $f$ in Lemma 5.6. ☐

We now show that the solution $\widehat{Y}$ has the desired spectral property.

THEOREM 5.8. [illegible] $\mu > 0$ [illegible] $\widehat{Y}$ [illegible] (5.3) [illegible] 5.7 [illegible] $\sigma(\widehat{A} - \widehat{B}\widehat{Y}) = \{-\lambda_{n+1}, \ldots, -\lambda_{n+m}\}$

[illegible] Notice that

$$\widehat{A} - \widehat{B}\widehat{Y} = A - BY - \eta v_2(p_1^T Y + p_2^T) - (B + \eta v_2 p_1^T)(Yv_2 - v_1)f^T.$$

For the matrix $W = \left[\begin{smallmatrix} W_{11} & W_{12} \\ W_{21} & W_{22}\end{smallmatrix}\right]$ in Lemma 5.5, we have

$$\begin{bmatrix} 1 & f^T \\ 0 & I \end{bmatrix} W \begin{bmatrix} 1 & f^T \\ 0 & I \end{bmatrix}^{-1} = \begin{bmatrix} 1 & f^T \\ 0 & I \end{bmatrix} W \begin{bmatrix} 1 & -f^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} \eta & 0 \\ W_{21} & -(\widehat{A} - \widehat{B}\widehat{Y}) \end{bmatrix},$$

where we have used Lemma 5.6. Therefore, the eigenvalues of $\widehat{A} - \widehat{B}\widehat{Y}$ are $-\lambda_{n+1}, \ldots, -\lambda_{n+m}$ by Lemma 5.5. ☐

The case $\mu = 0$ has to be treated separately. Since $Yv_2 = v_1$, the matrix $W$ in Lemma 5.5 does not have a left eigenvector of the form $(1, f^T)$ corresponding to the eigenvalue $\eta$. In this case, we need to assume $p_1 > 0$. Actually, it is advisable in general to use a vector $p$ with $p_1 > 0$ also in the case $\mu > 0$, since this choice of $p$ guarantees that the matrix $\widehat{Y}$ is bounded independently of the nearness to null recurrence (as can be seen in the proof of the following theorem). In the next section, however, we will use a vector $p$ without this assumption for a special class of the NARE (1.1). There, the boundedness of $\widehat{Y}$ independent of the nearness to null recurrence will be guaranteed in another way.

THEOREM 5.9. [illegible] $\mu = 0$ [illegible] $p_1 > 0$ [illegible] (5.3) [illegible] $\widehat{Y}$ [illegible] $\sigma(\widehat{A} - \widehat{B}\widehat{Y}) = \{-\lambda_{n+1}, \dots, -\lambda_{n+m}\}$

[illegible]. We use a continuity argument similar to the one used in [10] when a shift technique in [16] is used for null recurrent quasi-birth-death problems. We introduce the irreducible singular $M$-matrix

$$M(k) = \begin{bmatrix} D(k) & -C(k) \\ -B(k) & A(k) \end{bmatrix} = \begin{bmatrix} D & -C \\ -(1+\frac{1}{k})B & (1+\frac{1}{k})A \end{bmatrix}$$

($k = 1, 2, \dots$). The left and right eigenvectors of $M(k)$ corresponding to the zero eigenvalue are given by

$$u_1(k) = u_1, \quad u_2(k) = \left(1 + \frac{1}{k}\right)^{-1} u_2, \quad v_1(k) = v_1, \quad v_2(k) = v_2.$$

Thus, the NARE corresponding to $M(k)$

$$ZCZ - ZD - A(k)Z + B(k) = 0 \tag{5.4}$$

is positive recurrent since $u_1(k)^T v_1(k) > u_2(k)^T v_2(k)$. Let $Y(k)$ be the minimal nonnegative solution of the dual equation of (5.4). Then $Y(k)v_2 < v_1$ and in particular the sequence $\{Y(k)\}$ is bounded. When $M$ is replaced by $M(k)$, we have a matrix $W(k)$ corresponding to the matrix $W$ in Lemma 5.5. Let $(1, f(k)^T)$ be the left eigenvector of $W(k)$ corresponding to the eigenvalue $\eta$. Now, $\widehat{Y}(k) = Y(k) + (Y(k)v_2 - v_1)f(k)^T$ is a solution of the dual equation of

$$Z\widehat{C}Z - Z\widehat{D} - \widehat{A}(k)Z + \widehat{B}(k) = 0,$$

where $\widehat{A}(k) = A(k) - \eta v_2 p_2^T$ and $\widehat{B}(k) = B(k) + \eta v_2 p_1^T$, and the eigenvalues of $\widehat{A}(k) - \widehat{B}(k)\widehat{Y}(k)$ are those of $A(k) - B(k)Y(k)$. We need to show that the sequence $\{\widehat{Y}(k)\}$ is bounded. Since $(1, f(k)^T)W(k) = \eta(1, f(k)^T)$, we have

$$\begin{aligned} \eta &= \eta p_1^T(v_1 - Y(k)v_2) + f(k)^T(B(k) + \eta v_2 p_1^T)(v_1 - Y(k)v_2) \\ &\geq f(k)^T(\eta v_2 p_1^T)(v_1 - Y(k)v_2) \end{aligned}$$

and thus

$$f(k)^T(v_2 p_1^T)(v_1 - Y(k)v_2) = p_1^T(v_1 - Y(k)v_2)f(k)^T v_2 \leq 1.$$

Since $p_1, v_2 > 0$, $\{(v_1 - Y(k)v_2)f(k)^T\}$ is bounded and thus $\{\widehat{Y}(k)\}$ is bounded. Let $\widehat{Y}$ be any limit point of the sequence $\{\widehat{Y}(k)\}$. Then the eigenvalues of $\widehat{A} - \widehat{B}\widehat{Y}$ are those of $A - BY$ since $\lim Y(k) = Y$ by Theorem 3.3 of [12]. $\square$

When $\mu = 0$, the matrix $H$ has two zero eigenvalues. The above shift technique moves one zero eigenvalue to a positive number. We may use a [illegible] to move the other zero eigenvalue to a negative number. Recall that $Hv = 0$, where $v = \left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]$, and $w^T H = 0$, where $w = \left[\begin{smallmatrix} u_1 \\ -u_2 \end{smallmatrix}\right]$. We define the matrix

$$\overline{H} = H + \eta v p^T + \xi q w^T, \tag{5.5}$$

where $\eta > 0$, $\xi < 0$, $p$ and $q$ are such that $p^T v = q^T w = 1$. Since $v$ and $w$ are orthogonal vectors, the double-shift moves one zero eigenvalue to $\eta$ and the other to $\xi$. Indeed, the eigenvalues of $\widetilde{H} = H + \xi q w^T$ are those of $\widetilde{H}^T = H^T + \xi w q^T$, which are the eigenvalues of $H$ except that one zero eigenvalue is replaced by $\xi$, by Lemma 5.3. Also, the eigenvalues of $\overline{H} = \widetilde{H} + \eta v p^T$ are the eigenvalues of $\widetilde{H}$ except that the remaining zero eigenvalue is replaced by $\eta$, again by Lemma 5.3.

From $\overline{H}$ we may define a new Riccati equation

$$Z\overline{C}Z - Z\overline{D} - \overline{A}Z + \overline{B} = 0. \tag{5.6}$$

As before, the minimal nonnegative solution $X$ of (1.1) is a solution of (5.6) such that $\sigma(\overline{D} - \overline{C}X) = \{\eta, \lambda_1, \dots, \lambda_{n-1}\}$. However, it seems very difficult to determine the existence of a solution $\overline{Y}$ of the dual equation of (5.6) such that $\sigma(\overline{A} - \overline{B}\,\overline{Y}) = \{-\xi, -\lambda_{n+2}, \dots, -\lambda_{n+m}\}$. We will not investigate the double-shift any further in this paper.

## 6. The doubling algorithm applied to the shifted equation.

In this section we assume that $M$ is an irreducible singular $M$-matrix and $\mu \geq 0$. We will show that the doubling algorithm applied to (5.3) converges faster (if no breakdown occurs) than the doubling algorithm applied to (1.1). The applicability of the SDA algorithm to the shifted equation (5.3) is still a work in progress, but we will prove that no breakdown occurs under suitable assumptions on the matrix $\widehat{M}$.

### 6.1. Convergence properties.

By Theorems 5.4, 5.8, and 5.9, the matrices $X$ and $\widehat{Y}$ are such that

$$\widehat{H}\begin{bmatrix} I \\ X \end{bmatrix} = \begin{bmatrix} I \\ X \end{bmatrix}(\widehat{D} - \widehat{C}X), \quad \widehat{H}\begin{bmatrix} \widehat{Y} \\ I \end{bmatrix} = \begin{bmatrix} \widehat{Y} \\ I \end{bmatrix}(-(\widehat{A} - \widehat{B}\widehat{Y})), \tag{6.1}$$

where $\sigma(\widehat{D} - \widehat{C}X) = \{\lambda_1, \dots, \lambda_{n-1}, \eta\}$, $\sigma(\widehat{A} - \widehat{B}\widehat{Y}) = \{-\lambda_{n+1} \dots, -\lambda_{n+m}\}$. Recall that we need to assume $p_1 > 0$ for the vector $p$ used in the shift technique when $\mu = 0$, to get the second equation in (6.1).

We apply the Cayley transform with $\gamma > 0$ to each of the equations in (6.1), thus obtaining

$$\begin{aligned} (\widehat{H} - \gamma I)\begin{bmatrix} I \\ X \end{bmatrix} &= (\widehat{H} + \gamma I)\begin{bmatrix} I \\ X \end{bmatrix}\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X), \\ (\widehat{H} - \gamma I)\begin{bmatrix} \widehat{Y} \\ I \end{bmatrix}\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y}) &= (\widehat{H} + \gamma I)\begin{bmatrix} \widehat{Y} \\ I \end{bmatrix}. \end{aligned} \tag{6.2}$$

We then proceed as in section 3, with (3.3) and (3.7) replaced by the equations in (6.2). Assuming that no breakdown occurs, the doubling algorithm generates the sequences of matrices $\{\widehat{E}_k\}, \{\widehat{F}_k\}, \{\widehat{G}_k\}, \{\widehat{H}_k\}$. We prove the following convergence result.

THEOREM 6.1. [illegible] $M$ [illegible] $M$ [illegible] $\mu \geq 0$ [illegible] $\{\widehat{E}_k\}, \{\widehat{F}_k\}, \{\widehat{G}_k\}, \{\widehat{H}_k\}$ [illegible] (5.3) [illegible] $\{\widehat{H}_k\}$ $(\{\widehat{G}_k\})$ [illegible] $X$ $(\widehat{Y})$ [illegible]

$$\limsup_{k\to\infty} \sqrt[2^k]{\|\widehat{H}_k - X\|} \leq \rho(\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))\rho(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y})), \tag{6.3}$$

$$\limsup_{k\to\infty} \sqrt[2^k]{\|\widehat{G}_k - \widehat{Y}\|} \leq \rho(\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))\rho(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y})). \tag{6.4}$$

[illegible] As in section 4, we have

$$X - \widehat{H}_k = (I - \widehat{H}_k\widehat{Y})(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y}))^{2^k} X (\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))^{2^k}, \tag{6.5}$$

$$\widehat{Y} - \widehat{G}_k = (I - \widehat{G}_k X)(\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))^{2^k} \widehat{Y} (\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y}))^{2^k}. \tag{6.6}$$

We prove (6.3). The proof of (6.4) is similar. By writing the $\widehat{H}_k$ on the right-hand side of (6.5) as $X - (X - \widehat{H}_k)$, we find

$$\begin{aligned} &(X - \widehat{H}_k)\left(I - \widehat{Y}(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y}))^{2^k} X (\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))^{2^k}\right) \\ &\qquad = (I - X\widehat{Y})(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y}))^{2^k} X (\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))^{2^k}. \end{aligned} \tag{6.7}$$

Note that

$$\begin{aligned} &\sigma(\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X)) = \{\mathcal{C}_\gamma(\lambda_1), \dots, \mathcal{C}_\gamma(\lambda_{n-1}), \mathcal{C}_\gamma(\eta)\}, \\ &\sigma(\mathcal{C}_\gamma(D - CX)) = \{\mathcal{C}_\gamma(\lambda_1), \dots, \mathcal{C}_\gamma(\lambda_{n-1}), \mathcal{C}_\gamma(0)\}, \\ &\sigma(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y})) = \sigma(\mathcal{C}_\gamma(A - BY)) = \{\mathcal{C}_\gamma(-\lambda_{n+1}), \dots, \mathcal{C}_\gamma(-\lambda_{m+n})\}. \end{aligned}$$

By Theorem 2.2 and the property of the Cayley transform, we have

$$\rho(\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X)) < \rho(\mathcal{C}_\gamma(D - CX)) = 1, \quad \rho(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y})) = \rho(\mathcal{C}_\gamma(A - BY)) \leq 1. \tag{6.8}$$

Thus $I - \widehat{Y}(\mathcal{C}_\gamma(\widehat{A} - \widehat{B}\widehat{Y}))^{2^k} X (\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))^{2^k}$ converges to $I$ in (6.7), and (6.3) follows immediately. □

Recall that the sequence $\{H_k\}$ generated in section 3 is such that

$$\limsup_{k\to\infty} \sqrt[2^k]{\|H_k - X\|} \leq \rho(\mathcal{C}_\gamma(D - CX))\rho(\mathcal{C}_\gamma(A - BY)).$$

In view of (6.8), the convergence of the doubling algorithm applied to the shifted equation is faster than the convergence of the same algorithm applied to the original equation. As we will show in the numerical experiments, the number of steps necessary for convergence can decrease dramatically by using the shift technique. In particular, when $\mu = 0$, the SDA algorithm applied to the shifted equation still has quadratic convergence. According to the results in [12], the shift equation is also better conditioned than the original equation.

At this point it must be specified which is the best choice for the parameter $\eta$ in terms of the speed of convergence. In fact, the fastest convergence is expected when $\rho(\mathcal{C}_\gamma(\widehat{D} - \widehat{C}X))$ is minimal, i.e., when $|\mathcal{C}_\gamma(\eta)| \leq \max\{|\mathcal{C}_\gamma(\lambda_i)|, i = 1, \dots, n-1\}$. This happens, for instance, if $\eta = \gamma$ (i.e., $\mathcal{C}_\gamma(\eta) = 0$).

**6.2. Applicability for a special class of NARE.** We can prove the applicability of the SDA algorithm to the shifted equation for a special class of the NARE (1.1) and for proper choices of $\eta, p$, and $\gamma$.

The special class consists of equations for which either $C$ has at least one positive column or $D$ has at least one column with no zero entries.

For ease of notation, we assume without loss of generality [12] that $v = e$, the vector of ones. We define $r_1 \in \mathbb{R}^n$ and $r_2 \in \mathbb{R}^m$ by

$$(r_1)_j = \min_{1\le i\le n, i\ne j} |d_{ij}|, \quad j = 1, \ldots, n, \quad (r_2)_j = \min_{1\le i\le n} c_{ij}, \quad j = 1, \ldots, m,$$

and define $r \geq 0$ by $r^T = (r_1^T, r_2^T)$. We have $r \neq 0$ for each equation in the special class, and we use the shift (5.2) with $\eta = r^T e$ and $p = r/r^T e$.

Clearly, the matrix $\widehat{M}$ is a $Z$-matrix. It is interesting to note that $\widehat{M}$ is also an $M$-matrix. This follows from the fact that

$$\left(u_1^T, u_2^T\right) \widehat{M} = \eta \left(\mu p_1^T, \mu p_2^T\right) = \mu \left(r_1^T, r_2^T\right) \geq 0.$$

Recall that $\widehat{M}$ is nonsingular when $\mu > 0$ and is singular when $\mu = 0$. It should be noted that $\widehat{M}$ may be a [illegible] singular $M$-matrix. As a simple example, we have $\widehat{M} = \left[\begin{smallmatrix} 2 & 0 \\ -2 & 0 \end{smallmatrix}\right]$ when $M = \left[\begin{smallmatrix} 1 & -1 \\ -1 & 1 \end{smallmatrix}\right]$.

In view of the results in [15], the doubling algorithm can be applied to the NARE corresponding to $\widehat{M}$ with $\gamma > \max\{\max(a_{ii}-(r_2)_i), \max(d_{ii}+(r_1)_i)\}$, and the matrices to be inverted in the algorithm are all nonsingular $M$-matrices. The results in [15] are stated for nonsingular $M$-matrices, but are easily seen to be true also for singular $M$-matrices. By Theorem 3.2 we can also take $\gamma = \max\{\max(a_{ii}-(r_2)_i), \max(d_{ii}+(r_1)_i)\}$ when $\widehat{M}$ is irreducible.

For the convergence analysis of the doubling algorithm applied to the matrix $\widehat{M}$, we need to prove the existence of $\widehat{Y}$ with the property in (6.1). By our definition of the vector $p$, we no longer have $p_1 > 0$ unless all off-diagonal elements of $D$ are negative. Therefore, we need a different proof for the existence of $\widehat{Y}$ with the property in (6.1), for the case $\mu = 0$ (see Theorem 5.9).

For the proof, we need to assume that the $M$-matrix $\widehat{M}$ is such that $I \otimes \widehat{A} + \widehat{D}^T \otimes I$ is a nonsingular $M$-matrix, where $\otimes$ is the Kronecker product. This is an additional assumption only when $\mu = 0$ and $\widehat{M}$ is reducible, and it is guaranteed by the assumption in the next result.

LEMMA 6.2. [illegible] (1.1) [illegible] $A$ [illegible] $B$ [illegible] $I \otimes \widehat{A} + \widehat{D}^T \otimes I$ [illegible] $M$[illegible]

[illegible]. Note that

$$(D - r_2^T eI)e - (C - er_2^T)e = De - Ce = 0.$$

So $D - r_2^T eI$ is an $M$-matrix. Now,

$$\begin{aligned} I \otimes \widehat{A} + \widehat{D}^T \otimes I &\geq I \otimes (A - er_2^T) + (D - r_2^T eI)^T \otimes I + r_2^T eI \otimes I \\ &= I \otimes (A - er_2^T + r_2^T eI) + (D - r_2^T eI)^T \otimes I. \end{aligned}$$

Note that $(A - er_2^T + r_2^T eI)e = Ae = Be \geq 0$ and that $Be \neq 0$. If $A$ is irreducible, then $A - er_2^T + r_2^T eI$ is irreducible. If $B$ has no zero rows, then $Be > 0$. In either case, $A - er_2^T + r_2^T eI$ is a nonsingular $M$-matrix. Therefore, $I \otimes \widehat{A} + \widehat{D}^T \otimes I$ is also a nonsingular $M$-matrix. $\square$

We now prove the existence of $\widehat{Y}$ with the property in (6.1), assuming that $I \otimes \widehat{A} + \widehat{D}^T \otimes I$ is a nonsingular $M$-matrix for the special class of the NARE (1.1). First, we note that

$$Y\widehat{B}Y - Y\widehat{A} - \widehat{D}Y + \widehat{C} = \eta(Ye - e)(p_1^T Y + p_2^T) \le 0.$$

By Theorem 2.3 of [8], there is a minimal $\widehat{Y}$, $0 \le \widehat{Y} \le Y$, such that

$$\widehat{Y}\widehat{B}\widehat{Y} - \widehat{Y}\widehat{A} - \widehat{D}\widehat{Y} + \widehat{C} = 0.$$

Note that

$$\widehat{H}\begin{bmatrix} I & \widehat{Y} \\ X & I \end{bmatrix} = \begin{bmatrix} I & \widehat{Y} \\ X & I \end{bmatrix}\begin{bmatrix} \widehat{D} - \widehat{C}X & 0 \\ 0 & -(\widehat{A} - \widehat{B}\widehat{Y}) \end{bmatrix}.$$

When $\mu > 0$, we have $Xe = e$ and $\widehat{Y}e < e$ and the matrix

$$\begin{bmatrix} I & \widehat{Y} \\ X & I \end{bmatrix}$$

is nonsingular. So the eigenvalues of $\widehat{A} - \widehat{B}\widehat{Y}$ are $-\lambda_{n+1}, \ldots, -\lambda_{n+m}$. By a continuity argument, the eigenvalues of $\widehat{A} - \widehat{B}\widehat{Y}$ are also $-\lambda_{n+1}, \ldots, -\lambda_{n+m}$ when $\mu = 0$.

**7. Numerical experiments.** We compare the numerical behavior of the SDA algorithm applied to (1.1) and to the shifted equation (5.3), when $\mu \ge 0$. Recall that the case $\mu < 0$ is easily reduced to the case $\mu > 0$ through Lemma 5.1.

The numerical experiments are performed by using MATLAB; the stopping condition is $\min\{\|E_k\|_1, \|F_k\|_1\} < 10^{-15}$.

We take $\gamma = \max\{\max a_{ii}, \max d_{ii}\}$ for the Cayley transform, as suggested by Theorem 4.4. For the shift technique, we take $\eta = \gamma$ and $p = e/v^T e$, where $e$ is the vector (of suitable size) with all components equal to 1.

TEST 7.1 (see [8]). [illegible] $M$ [illegible] $Me = 0$. To construct $M$, we generate $R$, a $100 \times 100$ random matrix, and define $M = \mathrm{diag}(Re) - R$. The matrices $A, B, C$, and $D$ are $50 \times 50$.

We generate 5 different matrices $M$ in this way, each with $\mu > 0$. In Table 7.1 we report the number of iterations and the relative residual, defined as

$$\mathrm{res} = \frac{\|XCX - XD - AX + B\|_1}{\|XCX\|_1 + \|XD\|_1 + \|AX\|_1 + \|B\|_1}.$$

As one can see, the number of steps applied to the shifted equation is smaller, while the residual error remains roughly the same ($u \approx 2.2 \times 10^{-16}$ is the unit roundoff).

TABLE 7.1
*SDA applied to original and shifted NARE.*

| | SDA | | SDA applied to shifted NARE | |
|---|---|---|---|---|
| | iter | res/err | iter | res/err |
| Test 1 | 12–13 | res=$1.1u$–$2.1u$ | 5 | res=$1.2u$–$1.7u$ |
| Test 2 | 33 | err=$1.6 \times 10^{-9}$ | 5 | err=$2.2 \times 10^{-16}$ |
| Test 3 | 18 | err=$3.5 \times 10^{-13}$ | 4 | err=$2.3 \times 10^{-13}$ |

Test 7.2 (see [2, Example 1]). [illegible]. Let

$$M = \begin{bmatrix} 0.003 & -0.001 & -0.001 & -0.001 \\ -0.001 & 0.003 & -0.001 & -0.001 \\ -0.001 & -0.001 & 0.003 & -0.001 \\ -0.001 & -0.001 & -0.001 & 0.003 \end{bmatrix},$$

where $D$ is a $2 \times 2$ matrix. The minimal positive solution is $X = \frac{1}{2}E_{2,2}$, where $E_{m,n}$ is the $m \times n$ matrix having all entries equal to 1.

In this case the SDA algorithm shows linear convergence while the SDA applied to the shifted equation has quadratic convergence. Indeed, as reported in Table 7.1 the number of steps decreases dramatically. Since the solution is explicitly known, we have compared the absolute error, defined as the 1-norm of the difference between the exact and the computed solution, obtained with both methods. Observe that the solution computed without performing the shift is much less accurate than the one obtained by applying the shift. This phenomenon is to be expected in view of the theoretical results in [12].

Test 7.3 (see [2, Example 3]). [illegible]. In this example $A = \mathrm{diag}(0.018\, E_{2,1})$, $D = \mathrm{diag}(180.002\, E_{18,1}) - 10\, E_{18,18}$, $B = 0.001\, E_{2,18}$, and $C = B^T$. The solution is known to be $\frac{1}{18}E_{2,18}$. The results are shown in Table 7.1; the reduction of the number of iterations for the shifted equation is significant.

In all our experiments, the approximations to $X$ obtained by the SDA algorithm with the shift technique are more accurate than those obtained without the shift technique, for the same number of iterations. The approximations obtained by the shift technique converge to the positive solution $X$. However, when $X$ has very small positive elements, it is possible that at the time of termination of the algorithm the approximation obtained has some very small negative elements. We have not yet encountered such situations, but if those situations do occur, we can simply reset those small negative numbers to 0, indicating that the actual values are very small positive numbers.

## REFERENCES

[1] B. D. O. Anderson, *Second-order convergent algorithms for the steady-state Riccati equation*, Internat. J. Control, 28 (1978), pp. 295–306.

[2] N. G. Bean, M. M. O'Reilly, and P. G. Taylor, *Algorithms for return probabilities for stochastic fluid flows*, Stoch. Models, 21 (2005), pp. 149–184.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, PA, 1994.

[4] D. A. Bini, B. Iannazzo, G. Latouche, and B. Meini, *On the solution of algebraic Riccati equations arising in fluid queues*, Linear Algebra Appl., 413 (2006), pp. 474–494.

[5] C.-Y. Chiang and W.-W. Lin, *A Structured Doubling Algorithm for Nonsymmetric Algebraic Riccati Equations (A Singular Case)*, Technical report, NCTS, National Taiwan University, Taiwan, available online at http://math.cts.nthu.edu.tw/Mathematics/preprints/prep2006-7-001.pdf.

[6] S. Fital and C.-H. Guo, *Convergence of the solution of a nonsymmetric matrix Riccati differential equation to its stable equilibrium solution*, J. Math. Anal. Appl., 318 (2006), pp. 648–657.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, MD, 1989.

[8] C.-H. Guo, *Nonsymmetric algebraic Riccati equations and Wiener–Hopf factorization for M-matrices*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 225–242.

[9] C.-H. Guo, *A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation*, Linear Algebra Appl., 357 (2002), pp. 299–302.

[10] C.-H. Guo, *Comments on a shifted cyclic reduction algorithm for quasi-birth-death problems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1161–1166.

[11] C.-H. Guo, *Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models*, J. Comput. Appl. Math., 192 (2006), pp. 353–373.

[12] C.-H. Guo and N. J. Higham, *Iterative solution of a nonsymmetric algebraic Riccati equation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 396–412.

[13] C.-H. Guo and A. J. Laub, *On the iterative solution of a class of nonsymmetric algebraic Riccati equations*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 376–391.

[14] X.-X. Guo and Z.-Z. Bai, *On the minimal nonnegative solution of nonsymmetric algebraic Riccati equation*, J. Comput. Math., 23 (2005), pp. 305–320.

[15] X.-X. Guo, W.-W. Lin, and S.-F. Xu, *A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation*, Numer. Math., 103 (2006), pp. 393–412.

[16] C. He, B. Meini, and N. H. Rhee, *A shifted cyclic reduction algorithm for quasi-birth-death problems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 673–691.

[17] R. A. Horn and S. Serra-Capizzano, *Canonical and Standard Forms for Certain Rank One Perturbations and an Application to the (Complex) Google Pageranking Problem*, Internet Math., to appear.

[18] J. Juang, *Existence of algebraic matrix Riccati equations arising in transport theory*, Linear Algebra Appl., 230 (1995), pp. 89–100.

[19] J. Juang and W.-W. Lin, *Nonsymmetric algebraic Riccati equations and Hamiltonian-like matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 228–243.

[20] L.-Z. Lu, *Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 679–685.

[21] C. D. Meyer, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.

[22] V. Ramaswami, *Matrix analytic methods for stochastic fluid flows*, in Proceedings of the Sixteenth International Teletraffic Congress, New York, Elsevier Science B. V., Edinburgh, 1999, pp. 1019–1030.

[23] L. C. G. Rogers, *Fluid models in queueing theory and Wiener–Hopf factorization of Markov chains*, Ann. Appl. Probab., 4 (1994), pp. 390–413.

[24] L. C. G. Rogers and Z. Shi, *Computing the invariant law of a fluid model*, J. Appl. Probab., 31 (1994), pp. 885–896.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1101–1119

# BLOCK DIAGONAL AND SCHUR COMPLEMENT PRECONDITIONERS FOR BLOCK-TOEPLITZ SYSTEMS WITH SMALL SIZE BLOCKS*

WAI-KI CHING†, MICHAEL K. NG‡, AND YOU-WEI WEN§

**Abstract.** In this paper we consider the solution of Hermitian positive definite block-Toeplitz systems with small size blocks. We propose and study block diagonal and Schur complement preconditioners for such block-Toeplitz matrices. We show that for some block-Toeplitz matrices, the spectra of the preconditioned matrices are uniformly bounded except for a fixed number of outliers where this fixed number depends only on the size of the block. Hence, conjugate gradient type methods, when applied to solving these preconditioned block-Toeplitz systems with small size blocks, converge very fast. Recursive computation of such block diagonal and Schur complement preconditioners is considered by using the nice matrix representation of the inverse of a block-Toeplitz matrix. Applications to block-Toeplitz systems arising from least squares filtering problems and queueing networks are presented. Numerical examples are given to demonstrate the effectiveness of the proposed method.

**Key words.** block-Toeplitz matrix, block diagonal, Schur complement, preconditioners, recursion

**AMS subject classifications.** 65F10, 65N20

**DOI.** 10.1137/S0895479803428230

## 1. Introduction.

In this paper we consider the solution of a Hermitian positive definite block-Toeplitz (BT) system with small size blocks

$$A_{n,m}X = B, \tag{1.1}$$

where $X$ and $B$ are $mn$-by-$m$ matrices and

$$A_{n,m} = \begin{pmatrix} A_0 & A_{-1} & \cdots & A_{1-n} \\ A_1 & A_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_{-1} \\ A_{n-1} & \cdots & A_1 & A_0 \end{pmatrix},$$

where each $A_j$ is an $m$-by-$m$ matrix with $A_j = A_{-j}^*$ and $m$ is much smaller than $n$. Here "$*$" denotes the conjugate transpose. This kind of linear system arises from many applications such as the multichannel least squares filtering in time series [26], signal and image processing [20], and queueing system [11]. We will discuss these

---

*Received by the editors December 12, 2003; accepted for publication (in revised form) by P. C. Hansen April 17, 2007; published electronically November 9, 2007.
http://www.siam.org/journals/simax/29-4/42823.html

†Advanced Modeling and Applied Computing Laboratory and Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong (wkc@maths.hku.hk). The research of this author was supported in part by Hong Kong Research Grants Council grant HKU 7126/02P, HKU CRCG grants 10204436 and 10205105, Hung Hing Ying Physical Sciences Research Fund, and HKU Strategic Research Theme Fund on Computational Physics and Numerical Methods.

‡Corresponding author. Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (mng@math.hkbu.edu.hk). The research of this author was supported in part by Hong Kong Research Grants Council grants 7035/04P, 7035/05P, and HKBU FRGs.

§Faculty of Science, South China Agricultural University, Wushan, Guangzhou, P. R. China (wenyouwei@gmail.com).

applications, in particular the least squares filtering problems and queueing networks, in section 5.

Recent research on using the preconditioned conjugate gradient method as an iterative method for solving $n$-by-$n$ Toeplitz systems has received much attention.One of the more important results of this methodology is that the complexity of solving a large class of Toeplitz systems can be reduced to $O(n \log n)$ operations provided that a suitable preconditioner is chosen under certain conditions on the Toeplitz matrix [7]. Circulant preconditioners [3, 4, 8, 9, 10, 17, 25, 30, 33], banded-Toeplitz preconditioners [5], and multigrid methods [6, 12] have been proposed and analyzed. In these papers, the diagonals of the Toeplitz matrix are assumed to be the Fourier coefficients of a certain generating function.

In the literature, there are some papers [18, 21, 22, 27, 28, 29, 31] which discuss iterative BT solvers. In [21, 28, 29], the authors considered $n$-by-$n$ BT matrices with $m$-by-$m$ blocks generated by a Hermitianmatrix-valued generating function and analyzed the associated problem of preconditioning using preconditioners which generated nonnegative definite,not essentially singular, matrix-valued functions. In [18, 22, 27], the authors considered block-Toeplitz–Toeplitz-block matrices and studied block band-Toeplitz preconditioners. In [31], multigrid methods were applied to solving block-Toeplitz–Toeplitz-block systems. In the above methods, the underlying generating functions are assumed to be known in order to construct the preconditioners.

In this paper, we also consider BT matrices $A_{n,m}$ generated by a matrix-valued function

$$F_m(\theta) = [f_{u,v}(\theta)]_{1 \le u,v \le m},$$

where $f_{u,v}(\theta)$ are $2\pi$-periodic functions. Under this assumption, the block $A_j$ of $A_{n,m}$ is given by

$$A_j = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_m(\theta) e^{-\mathrm{i}j\theta} d\theta.$$

When $F_m(\theta)$ is nonnegative definite and not essentially singular, the associated BT matrix $A_{n,m}$ is positive definite [21, 28]. For such BT matrices, Serra [28] has investigated BT preconditioners and studied the spectral property of these preconditioned matrices. He proved that if the BT preconditioner is generated by $G_m(\theta)$, the generalized Rayleigh quotient, related to matrix functions $F_m(\theta)$ and $G_m(\theta)$, is contained in a set of the form $(c_1, c_2)$ with $0 < c_1$ and $c_2 < \infty$, then the preconditioned conjugate gradient (PCG) method requires only a constant number of iterations in order to solve, within a preassigned accuracy, the given BT system.

In [24], Ng, Sun, and Jin proposed to using recursive-based PCG methods for solving Toeplitz systems. The idea is to use a principal submatrix of a Toeplitz matrix as a preconditioner. The inverse of the preconditioner can be constructed recursively by using the Gohberg–Semencul formula. They have shown that this method is competitive with the method of circulant preconditioners. Based on this idea, the main aim of this paper is to study block diagonal and Schur complement preconditioners for BT systems. We note that there is a natural partitioning of the BT matrix in 2-by-2 blocks as follows:

$$A_{n,m} = \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} \end{pmatrix}. \tag{1.2}$$

Here $A^{(1,1)}$ and $A^{(2,2)}$ are the principal submatrices of $A_{n,m}$. They are also BT matrices generated by the same generating function of $A_{n,m}$. Therefore it is natural and important to examine if the corresponding system

$$\begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \tag{1.3}$$

can be solved efficiently by exploiting this partitioning. Here we consider preconditioning $A_{n,m}$ by a block diagonal matrix

$$B_{n,m} = \begin{pmatrix} A^{(1,1)} & 0 \\ 0 & A^{(2,2)} \end{pmatrix}.$$

Since both $A^{(1,1)}$ and $A^{(2,2)}$ are BT matrices generated by the same generating function $F_m(\theta)$, we particularly consider $B_{n,m}$ in the following form:

$$B_{n,m} = \begin{pmatrix} A_{n/2,m} & 0 \\ 0 & A_{n/2,m} \end{pmatrix}. \tag{1.4}$$

Here, without loss of generality, we may assume $n$ is even. We note that if $A_{n,m}$ is positive definite, then $B_{n,m}$ is also positive definite and the eigenvalues of the preconditioned matrix $B_{n,m}^{-1}A_{n,m}$ lie in the interval $(0, 2)$.

On the other hand, the Schur complement arises when we use a block factorization of (1.2). The linear system (1.3) becomes

$$\begin{pmatrix} I & 0 \\ A^{(2,1)}(A^{(1,1)})^{-1} & I \end{pmatrix} \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ 0 & S_{n,m} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix},$$

where

$$S_{n,m} = A^{(2,2)} - A^{(2,1)}(A^{(1,1)})^{-1}A^{(1,2)}.$$

We see that the method requires the formation of the Schur complement matrix. Therefore we consider approximating $S_{n,m}$ by $A^{(2,2)} = A_{n/2,m}$ and study the preconditioner of the form

$$\begin{aligned} C_{n,m} &= \begin{pmatrix} I & 0 \\ A^{(2,1)}(A^{(1,1)})^{-1} & I \end{pmatrix} \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ 0 & A^{(2,2)} \end{pmatrix} \\ &= \begin{pmatrix} A^{(1,1)} & A^{(1,2)} \\ A^{(2,1)} & A^{(2,2)} + A^{(2,1)}(A^{(1,1)})^{-1}A^{(1,2)} \end{pmatrix}. \end{aligned} \tag{1.5}$$

We note that if $A_{n,m}$ is positive definite, then $C_{n,m}$ is also positive definite and the eigenvalues of the preconditioned matrix $C_{n,m}^{-1}A_{n,m}$ are inside of the interval $(0, 1]$. In particular, there are at least $mn/2$ eigenvalues of the preconditioned matrix being equal to one. Our experimental results also show that the Schur-complement preconditioner is better than the block diagonal preconditioner. We remark that the main reason for discussing the block diagonal preconditioner is that it is needed for deriving the theory for the Schur-complement preconditioner.

The main result of this paper is that if the generating function $F_m(\theta)$ is Hermitian positive definite, and is spectrally equivalent to

$$G_m(\theta) = [g_{u,v}]_{1 \le u,v \le m},$$

where $g_{u,v}$ are trigonometric polynomials, then the spectra of the preconditioned matrices $B_{n,m}^{-1}A_{n,m}$ and $C_{n,m}^{-1}A_{n,m}$ are uniformly bounded except for a fixed number of outliers where the number of outliers depends only on $m$. Hence the conjugate gradient type methods, when applied to solving these preconditioned BT systems, converge very quickly, especially when $m$ is small.

The goal of this paper is to construct preconditioners that do not require matrix generating functions. We note that the construction of our preconditioners does not require the underlying matrix generating functions, while the preconditioners from [21, 28] require matrix generating functions. In the construction of our preconditioners, the inverse of BT matrix $A^{(1,1)}$ is required. Using the same idea in [24], we employ the Gohberg–Semencul formula to represent the form of the inverse of $A^{(1,1)}$ and apply a recursive method to construct the inverse of $A^{(1,1)}$. It is important to note that we do not directly use the Gohberg–Semencul formula to generate the solution of the original BT system.

We remark that the solution results are not accurate when the BT matrices are ill-conditioned. Indeed, we use the Gohberg–Semencul formula to generate an approximate inverse preconditioner and then use the PCG method with this preconditioner to compute the solution of the original system iteratively. Our numerical results indicate that the accuracy of the computed solutions using the proposed preconditioners is quite acceptable.

The outline of this paper is as follows. In section 2, we analyze the spectra of the preconditioned matrices. In section 3, we describe the recursive algorithms for block diagonal and Schur complement preconditioners. Numerical results are given in section 4 to illustrate the effectiveness of our approach. Finally, concluding remarks are given in section 5.

**2. Analysis of preconditioners.** In this section, we analyze the spectra of the preconditioned matrices $B_{m,n}^{-1}A_{n,m}$ and $C_{n,m}^{-1}A_{n,m}$.

We first note that since $A_{n,m}$ is positive definite, we have the following results, which are given in [1, pp. 374–377].

LEMMA 2.1. [illegible] $\mathbf{x}$ [illegible] $\mathbf{y}$ [illegible] $mn/2$ [illegible]

$$\gamma = \sup_{\mathbf{x},\mathbf{y}} \frac{\mathbf{x}^* A_{n/2,m}^{(1)} \mathbf{y}}{\sqrt{\mathbf{x}^* A_{n/2,m}\mathbf{x} \cdot \mathbf{y}^* A_{n/2,m}\mathbf{y}}}.$$

[illegible] $A_{n,m}$ [illegible] $\gamma < 1$ [illegible]

$$\gamma^2 = \sup_{\mathbf{y}} \frac{\mathbf{y}^* A_{n/2,m}^{2,1} A_{n/2,m}^{-1} A_{n/2,m}^{1,2} \mathbf{y}}{\mathbf{y}^* A_{n/2,m}\mathbf{y}}.$$

Using Lemma 2.1 and the assumption that $A_{n,m}$ is Hermitian and positive definite, we have the following results:

- The eigenvalues of the preconditioned matrix $B_{n,m}^{-1}A_{n,m}$ lie inside the interval $(0, 2)$. Also if $\mu$ is an eigenvalue of $B_{n,m}^{-1}A_{n,m}$, then $2-\mu$ is also an eigenvalue of $B_{n,m}^{-1}A_{n,m}$.
- The eigenvalues of $C_{n,m}^{-1}A_{n,m}$ are inside the interval $(0, 1]$. Moreover, at least $mn/2$ eigenvalues of $C_{n,m}^{-1}A_{n,m}$ are equal to 1.

We then show that the eigenvalues of $B_{n,m}^{-1}A_{n,m}$ and $C_{n,m}^{-1}A_{n,m}$ are uniformly bounded except for a fixed number of outliers for some generation functions $F_m(\theta)$. We first let

$$E_n(\theta) = [e_{u,v}(\theta)]_{1\le u,v\le n}, \quad \text{where} \quad e_{u,v}(\theta) = e^{-\mathrm{i}(u-v)\theta}.$$

The BT matrix $A_{n,m}$ can be expressed in terms of its generating function:

$$A_{n,m} = \frac{1}{2\pi}\int_{-\pi}^{\pi} E_n(\theta)\otimes F_m(\theta)d\theta. \tag{2.1}$$

Similarly, the block diagonal preconditioner can be expressed as follows:

$$B_{n,m} = \frac{1}{2\pi}\int_{-\pi}^{\pi} \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} \otimes F_m(\theta)d\theta. \tag{2.2}$$

We note that there exists a permutation matrix $P_{n,m}$ such that

$$P_{n,m}^* A_{n,m} P_{n,m} = \tilde{A}_{n,m} = \frac{1}{2\pi}\int_{-\pi}^{\pi} F_m(\theta)\otimes E_n(\theta)d\theta$$

and

$$P_{n,m}^* B_{n,m} P_{n,m} = \tilde{B}_{n,m} = \frac{1}{2\pi}\int_{-\pi}^{\pi} F_m(\theta)\otimes \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} d\theta.$$

It is clear that $\tilde{A}_{n,m}$ and $\tilde{B}_{n,m}$ are Toeplitz-block (TB) matrices, and the spectra of $A_{n,m}$ and $\tilde{A}_{n,m}$, and $B_{n,m}$ and $\tilde{B}_{n,m}$ are the same. Since the spectra of $B_{n,m}^{-1}A_{n,m}$ and $\tilde{B}_{n,m}^{-1}\tilde{A}_{n,m}$ are the same, it suffices to study the spectral properties of $\tilde{B}_{n,m}^{-1}\tilde{A}_{n,m}$.

We give the following two lemmas.

LEMMA 2.2. [illegible] $A = [a_{i,j}]_{1\le i,j\le m}$ [illegible] $B = [b_{i,j}]_{1\le i,j\le n}$ [illegible] $n$ [illegible] $m$ [illegible] $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$ [illegible] $Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$ [illegible]

$$\operatorname{vec}(X)^*(A\otimes B)\operatorname{vec}(Y) = \sum_{u=1}^{m}\sum_{v=1}^{m} a_{u,v}\mathbf{x}_u^* B\mathbf{y}_v \tag{2.3}$$

[illegible] $\operatorname{vec}(X) = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{pmatrix}$ [illegible] $\operatorname{vec}(Y) = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{pmatrix}$

LEMMA 2.3. [illegible] $\mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{pmatrix}$ [illegible] $\mathbf{x}_l = \begin{pmatrix} x_{(l-1)n+1} \\ x_{(l-1)n+2} \\ \vdots \\ x_{ln} \end{pmatrix}$ $(1 \le l \le m)$ $\mathbf{p}_1(\theta) = \begin{pmatrix} \check{\mathbf{p}}_{11}(\theta) \\ \check{\mathbf{p}}_{21}(\theta) \\ \vdots \\ \check{\mathbf{p}}_{m1}(\theta) \end{pmatrix}$ [illegible] $\check{\mathbf{p}}_{j1}(\theta) = \sum_{l=1}^{n'} x_{(j-1)n+l}e^{-\mathrm{i}(l-1)\theta}$ [illegible] $\mathbf{p}_2(\theta) = \begin{pmatrix} \check{\mathbf{p}}_{12}(\theta) \\ \check{\mathbf{p}}_{22}(\theta) \\ \vdots \\ \check{\mathbf{p}}_{m2}(\theta) \end{pmatrix}$ [illegible] $\check{\mathbf{p}}_{j2}(\theta) = e^{-\mathrm{i}n'\theta}\sum_{l=1}^{n-n'} x_{(j-1)n+n'+l}e^{-\mathrm{i}(l-1)\theta}$ [illegible] $A_{n,m}$ [illegible] $F_m(\theta)$ [illegible]

$$\mathbf{x}^*\tilde{B}_{n,m}\mathbf{x} = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\mathbf{p}_1(\theta)^*F_m(\theta)\overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)^*F_m(\theta)\overline{\mathbf{p}_2(\theta)}\right]d\theta \tag{2.4}$$

[illegible]

$$\mathbf{x}^*\tilde{A}_{n,m}\mathbf{x} = \mathbf{x}^*\tilde{B}_{n,m}\mathbf{x} + \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\mathbf{p}_1(\theta)^*F_m(\theta)\overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta)^*F_m(\theta)\overline{\mathbf{p}_1(\theta)}\right]d\theta. \tag{2.5}$$

[illegible] We construct $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m)$, i.e., $\mathbf{x} = \mathrm{vec}(X)$. Using Lemma 2.2, we obtain

$$\mathrm{vec}(X)^* \tilde{A}_{n,m} \mathrm{vec}(X) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{u=1}^{m} \sum_{v=1}^{m} f_{u,v}(\theta) \mathbf{x}_u^* E_n(\theta) \mathbf{x}_v d\theta \tag{2.6}$$

and

$$\mathrm{vec}(X)^* \tilde{B}_{n,m} \mathrm{vec}(X) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{u=1}^{m} \sum_{v=1}^{m} f_{u,v}(\theta) \mathbf{x}_u^* \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} \mathbf{x}_v d\theta. \tag{2.7}$$

We note that

$$\begin{aligned}
&\mathbf{x}_u^* \begin{pmatrix} E_{n/2}(\theta) & 0 \\ 0 & E_{n/2}(\theta) \end{pmatrix} \mathbf{x}_v \\
&= \sum_{j=1}^{n/2} x_{(u-1)n+j} \sum_{l=1}^{n/2} x_{(v-1)n+l} e_{jl}(\theta) + \sum_{j=n/2+1}^{n} x_{(u-1)n+j} \sum_{l=n/2+1}^{n} x_{(v-1)n+l} e_{jl}(\theta) \\
&= \sum_{j=1}^{n/2} x_{(u-1)n+j} e^{-\mathrm{i}(j-1)} \sum_{l=1}^{n/2} x_{(v-1)n+l} e^{\mathrm{i}(l-1)} \\
&\quad + \sum_{j=n/2+1}^{n} x_{(u-1)n+j} e^{-\mathrm{i}(j-1)} \sum_{l=n/2+1}^{n} x_{(v-1)n+l} e^{\mathrm{i}(l-1)} \\
&= \check{\mathbf{p}}_{u1}(\theta) \overline{\check{\mathbf{p}}_{v1}(\theta)} + \check{\mathbf{p}}_{u2}(\theta) \overline{\check{\mathbf{p}}_{v2}(\theta)}.
\end{aligned}$$

By using (2.7), one can obtain (2.4) directly. Similarly by using (2.6), (2.5) can also be derived. □

Next, we show that the eigenvalues of $B_{n,m}^{-1} A_{n,m}$ are uniformly bounded except for a fixed number of outliers when $F_m(\theta)$ is Hermitian positive definite and is spectrally equivalent to $G_m(\theta) = [g_{u,v}]_{1 \le u,v \le m}$, where $g_{u,v}$ are trigonometric polynomials. We remark that the fixed number of outliers depends on $m$.

THEOREM 2.4. [illegible] $F_m(\theta)$ [illegible] $F_m(\theta)$ [illegible] $G_m(\theta) = [g_{u,v}]_{1 \le u,v \le m}$ [illegible] $g_{u,v}$ [illegible] $s$ [illegible] $G_m(\theta)$ [illegible] $\alpha$ [illegible] $\beta$ $(\alpha < \beta)$ [illegible] $n$ [illegible] $n > 2s'$ $(s' = \lceil s/2 \rceil)$ [illegible] $2ms'$ [illegible] $\tilde{B}_{n,m}^{-1} \tilde{A}_{n,m}$ [illegible] $B_{n,m}^{-1} A_{n,m}$ [illegible] $[\alpha, \beta]$

[illegible] We note that there exist positive numbers $\gamma_1$ and $\gamma_2$ such that

$$0 < \gamma_1 \le \frac{\mathbf{y}^* F_m(\theta) \mathbf{y}}{\mathbf{y}^* G_m(\theta) \mathbf{y}} \le \gamma_2 \quad \forall \mathbf{y} \in \mathbb{R}^m, \ \forall \theta \in [0, 2\pi]. \tag{2.8}$$

We define the two sets $\Upsilon$ and $\Omega$ as follows:

$$\Upsilon = \{r : r = jn+n/2-s', jn+n/2-s'+1, \ldots, jn+n/2+s'-1 \text{ for } j = 0, 1, \ldots, m-1\}$$

and $\Omega = \left\{ \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{mn} \end{pmatrix} \mid z_k = 0 \text{ for } k \in \Upsilon \right\}$. We note that $\Omega$ is an $(mn - 2ms')$-dimensional subspace in $\mathbb{R}^{mn}$. It follows that for $\mathbf{x} \in \Omega$ and $\mathbf{p}_u(\theta)$ $(u = 1, 2)$ defined in Lemma 2.3, we have

$$\begin{aligned}
&\int_{-\pi}^{\pi} \mathbf{p}_1(\theta)^* G_m(\theta)\overline{\mathbf{p}_2(\theta)}d\theta = \int_{-\pi}^{\pi} \sum_{u=1}^{m}\sum_{v=1}^{m} \check{\mathbf{p}}_{u1}(\theta) f_{u,v}(\theta)\overline{\check{\mathbf{p}}_{v2}(\theta)}d\theta \\
&= \int_{-\pi}^{\pi} \sum_{u=1}^{m}\sum_{v=1}^{m} f_{u,v}(\theta)e^{\mathrm{i}n/2\theta}\sum_{j=1}^{n/2} x_{(u-1)n+j}e^{-\mathrm{i}(j-1)\theta}\sum_{j=1}^{n/2} x_{(v-1)n+n/2+j}e^{\mathrm{i}(j-1)\theta}d\theta \\
&= \sum_{u=1}^{m}\sum_{v=1}^{m}\int_{-\pi}^{\pi} f_{u,v}(\theta)e^{\mathrm{i}(2s'+1)\theta}\sum_{j=1}^{n/2-s'} x_{(u-1)n+j}e^{-\mathrm{i}j\theta} \\
&\cdot \sum_{j=1}^{n/2-s'} x_{(v-1)n+j+s'}e^{\mathrm{i}(n/2-s'-1+j)\theta}d\theta \;=\; 0 \qquad (2.9)
\end{aligned}$$

and

$$\begin{aligned}
&\int_{-\pi}^{\pi} \mathbf{p}_2(\theta)^* G_m(\theta)\overline{\mathbf{p}_1(\theta)}d\theta = \int_{-\pi}^{\pi} \sum_{u=1}^{m}\sum_{v=1}^{m} \check{\mathbf{p}}_{u2}(\theta)(\theta) f_{u,v}(\theta)\overline{\check{\mathbf{p}}_{v1}(\theta)(\theta)}d\theta \\
&= \int_{-\pi}^{\pi} \sum_{u=1}^{m}\sum_{v=1}^{m} f_{u,v}(\theta)e^{-\mathrm{n}/2\theta}\sum_{j=1}^{n/2} x_{(u-1)n+n/2+j}e^{-\mathrm{i}(j-1)\theta}\sum_{l=1}^{n/2} x_{(v-1)n+l}e^{\mathrm{i}(l-1)\theta}d\theta \\
&= \sum_{u=1}^{m}\sum_{v=1}^{m}\int_{-\pi}^{\pi} f_{u,v}(\theta)e^{-\mathrm{i}(2s'+1)\theta}\sum_{j=1}^{n/2-s'} x_{(u-1)n+n/2+s'+j}e^{-\mathrm{i}(n/2-s'-1+j)\theta} \\
&\cdot \sum_{l=1}^{n/2-s'} x_{(v-1)n+l}e^{\mathrm{i}j\theta}d\theta \;=\; 0. \qquad (2.10)
\end{aligned}$$

Since $F_m(\theta) - \gamma_1 G_m(\theta)$ is positive semidefinite, we have

$$\begin{aligned}
&\int_{-\pi}^{\pi} \mathbf{p}_1(\theta)^*[F_m - \gamma_1 G_m(\theta)](\theta)\overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)^*[F_m(\theta) - \gamma_1 G_m(\theta)]\overline{\mathbf{p}_2(\theta)}d\theta \\
&\geq \int_{-\pi}^{\pi} \mathbf{p}_1(\theta)^*[F_m - \gamma_1 G_m(\theta)](\theta)\overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta)^*[F_m(\theta) - \gamma_1 G_m(\theta)]\overline{\mathbf{p}_1(\theta)}d\theta. \qquad (2.11)
\end{aligned}$$

By using Lemma 2.3, (2.9), (2.10), and (2.11), we get

$$\begin{aligned}
&\left|\frac{\mathbf{x}^*\tilde{T}_{n,m}\mathbf{x} - \mathbf{x}^*\tilde{B}_{n,m}\mathbf{x}}{\mathbf{x}^*\tilde{B}_{n,m}\mathbf{x}}\right| = \frac{\left|\int_{-\pi}^{\pi}(\mathbf{p}_1(\theta)F_m(\theta)\overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta))F_m(\theta)\overline{(\mathbf{p}_1(\theta))}d\theta\right|}{\left|\int_{-\pi}^{\pi}(\mathbf{p}_1(\theta)F_m(\theta)\overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta))F_m(\theta)\overline{(\mathbf{p}_2(\theta))}d\theta\right|} \\
&= \frac{\left|\int_{-\pi}^{\pi} \mathbf{p}_1(\theta)[F_m(\theta) - \gamma_1 G_m(\theta)]\overline{\mathbf{p}_2(\theta)} + \mathbf{p}_2(\theta)[F_m(\theta) - \gamma_1 G_m(\theta)]\overline{\mathbf{p}_1(\theta)}d\theta\right|}{\left|\int_{-\pi}^{\pi} \mathbf{p}_1(\theta)F_m(\theta)\overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)F_m(\theta)\overline{\mathbf{p}_2(\theta)}d\theta\right|}. \\
&\leq \frac{\left|\int_{-\pi}^{\pi} \mathbf{p}_1(\theta)[F_m(\theta) - \gamma_1 G_m(\theta)]\overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)[F_m(\theta) - \gamma_1 G_m(\theta)]\overline{\mathbf{p}_2(\theta)}d\theta\right|}{\left|\int_{-\pi}^{\pi} \mathbf{p}_1(\theta)F_m(\theta)\overline{\mathbf{p}_1(\theta)} + \mathbf{p}_2(\theta)F_m(\theta)\overline{\mathbf{p}_2(\theta)}d\theta\right|} \\
&\leq 1 - \frac{\gamma_1}{\gamma_2} \quad \forall \mathbf{x} \in \Omega.
\end{aligned}$$

Therefore, we have

$$\alpha \equiv \frac{\gamma_1}{\gamma_2} \le \frac{\mathbf{x}^* \tilde{A}_{n,m}\mathbf{x}}{\mathbf{x}^* \tilde{B}_{n,m}\mathbf{x}} \le 2 - \frac{\gamma_1}{\gamma_2} \equiv \beta \quad \forall \mathbf{x} \in \Omega.$$

It implies that there are at most $2ms'$ eigenvalues of $\tilde{B}_{n,m}^{-1}\tilde{A}_{n,m}$ outside the interval $[\alpha, \beta]$. ☐

In [28], Serra explicitly constructed $G_m(\theta)$ by using eigendecomposition of $F_m(\theta)$:

$$F_m(\theta) = Q(\theta)^* \Lambda(\theta) Q(\theta),$$

where $\Lambda(\theta)$ is a diagonal matrix containing the eigenvalues $\lambda_j(F_m(\theta))$ $(j = 1, \dots, m)$ of $F_m(\theta)$. Suppose $\lambda_j(F_m(\theta))$ has a zero at $\theta_j$ of even order $\nu_j$. Then $G_m(\theta)$ is constructed in the following way:

$$G_m(\theta) = \sum_{j=1}^{m} Q(\theta_j)^* \Gamma(\theta) Q(\theta_j),$$

where $\Gamma(\theta)$ is a diagonal matrix with

$$[\Gamma(\theta)]_{kk} = \begin{cases} (2 - 2\cos(\theta))^{\nu_j/2}, & k = j, \\ 1 & \text{otherwise.} \end{cases}$$

It is clear that each entry of $G_m(\theta)$ is a polynomial. The largest degree of the polynomials in $G_m(\theta)$ depends on the orders of the zeros of the eigenvalues of $F_m(\theta)$. It has been shown that $F_m(\theta)$ is spectrally equivalent to $G_m(\theta)$; see, for instance, [28].

Similarly, we show that the eigenvalues of $C_{n,m}^{-1}A_{n,m}$ are uniformly bounded except for a fixed number of outliers, where this fixed number depends on $m$.

THEOREM 2.5. [illegible] $F_m(\theta)$ [illegible] $F_m(\theta)$ [illegible] $G_m(\theta) = [g_{u,v}]_{1 \le u,v \le m}$ [illegible] $g_{u,v}$ [illegible] $s$ [illegible] $G_m(\theta)$ [illegible] $\alpha$ [illegible] $\beta$ $(\alpha < \beta)$ [illegible] $n$ [illegible] $n > 2s'$ $(s' = \lceil s/2 \rceil)$ [illegible] $ms'$ [illegible] $\tilde{C}_{n,m}^{-1}\tilde{A}_{n,m}$ [illegible] $C_{n,m}^{-1}A_{n,m}$ [illegible] $[\alpha, \beta]$

[illegible] We note from (1.4) and (1.5) that

$$\det[B_{n,m}^{-1}(A_{n,m} - B_{n,m}) - \lambda I] = \det \begin{pmatrix} -\lambda I & A_{n/2,m}^{-1}A_{n,m}^{1,2} \\ A_{n/2,m}^{-1}A_{n,m}^{2,1} & -\lambda I \end{pmatrix} = 0$$

and

$$\det[C_{n,m}^{-1}(A_{n,m} - C_{n,m}) - \lambda I] = \det(-\lambda I)\det(A_{n/2,m}^{-1}A_{n,m}^{2,1}A_{n/2,m}^{-1}A_{n,m}^{1,2} - \lambda I) = 0.$$

Therefore, when the eigenvalues of $B_{n,m}^{-1}A_{n,m}$ are equal to $1 - \lambda$, the eigenvalues of $C_{n,m}^{-1}A_{n,m}$ are given by $1 - \lambda^2$. Using Theorem 2.4, we can find two positive numbers $\alpha = (\gamma_1/\gamma_2)^2$ and $\beta = 1$ such that the result holds. ☐

## 3. Recursive computation of $B_{n,m}^{-1}$ and $C_{n,m}^{-1}$.

In the previous section, we have shown that both $B_{n,m}$ and $C_{n,m}$ are good preconditioners for $A_{n,m}$. However, the inverses of $B_{n,m}$ and $C_{n,m}$ involve the inverse of $A_{n/2,m}$. The computational cost is still expensive. In this section, we present a recursive method to construct the preconditioners $B_{n,m}$ and $C_{n,m}$ efficiently.

We remark that the inverse of a Toeplitz matrix can be reconstructed by a low number of columns. Gohberg and Semencul [13] and Trench [32] showed that if the $(1,1)$st entry of the inverse of a Toeplitz matrix is nonzero, then the first and last columns of the inverse of the Toeplitz matrix are sufficient for this purpose. A nice matrix representation of the inverse, well known as the [illegible], was presented. In [16], an inversion formula was exhibited which works for every nonsingular Toeplitz matrix and uses the solutions of two equations (the so-called fundamental equations), where the right-hand side of one of them is a shifted column of the Toeplitz matrix. Later Ben-Artzi and Shalom [2], Labahn and Shalom [19], Ng, Rost, and Wen [23], and Heinig [15] studied the representation when the (1,1)st entry of the inverse of a Toeplitz matrix is zero. In [24], Ng, Sun, and Jin used the matrix representation of the inverse of a Toeplitz matrix to construct effective preconditioners for Toeplitz matrices.

For BT matrices, Gohberg and Heinig [14] also extended the Gohberg–Semencul formula to handle this case. It was shown that if $A_{n,m}$ is nonsingular, then the following equations are solvable:

$$A_{n,m}U^{(n)} = E^{(n)} \quad \text{and} \quad A_{n,m}V^{(n)} = F^{(n)} \tag{3.1}$$

with

$$U^{(n)} = \begin{pmatrix} U_1^{(n)} \\ U_2^{(n)} \\ \vdots \\ U_n^{(n)} \end{pmatrix}, \quad V^{(n)} = \begin{pmatrix} V_1^{(n)} \\ V_2^{(n)} \\ \vdots \\ V_n^{(n)} \end{pmatrix}, \quad E^{(n)} = \begin{pmatrix} I_m \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad F^{(n)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ I_m \end{pmatrix}.$$

Here $U_j^{(n)}$ and $V_j^{(n)}$ are $m$-by-$m$ matrices and $I_m$ is the identity matrix. Assuming that $U_1^{(n)}$ and $V_n^{(n)}$ are nonsingular, the inverse of $A_{n,m}$ can be expressed as follows:

$$A_{n,m}^{-1} = \Psi_{n,m}W_{n,m}\Psi_{n,m}^* - \Phi_{n,m}Z_{n,m}\Phi_{n,m}^*, \tag{3.2}$$

where $\Psi_{n,m}$ and $\Phi_{n,m}$ are $mn$-by-$mn$ lower triangular BT matrices given, respectively, by

$$\Psi_{n,m} = \begin{pmatrix} U_1^{(n)} & 0 & \cdots & 0 & 0 \\ U_2^{(n)} & U_1^{(n)} & 0 & & 0 \\ \vdots & U_2^{(n)} & U_1^{(n)} & 0 & \vdots \\ U_{n-1}^{(n)} & & \ddots & \ddots & 0 \\ U_n^{(n)} & U_{n-1}^{(n)} & \cdots & U_2^{(n)} & U_1^{(n)} \end{pmatrix}$$

and

$$\Phi_{n,m} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ V_1^{(n)} & 0 & 0 & & 0 \\ \vdots & V_1^{(n)} & 0 & 0 & \vdots \\ V_{n-2}^{(n)} & & \ddots & \ddots & 0 \\ V_{n-1}^{(n)} & V_{n-2}^{(n)} & \cdots & V_1^{(n)} & 0 \end{pmatrix}.$$

Moreover, $W_{n,m}$ and $Z_{n,m}$ are block diagonal matrices:

$$W_{n,m} = \begin{pmatrix} (U_1^{(n)})^{-1} & & & 0 \\ & (U_1^{(n)})^{-1} & & \\ & & \ddots & \\ 0 & & & (U_1^{(n)})^{-1} \end{pmatrix},$$

$$Z_{n,m} = \begin{pmatrix} (V_n^{(n)})^{-1} & & & 0 \\ & (V_n^{(n)})^{-1} & & \\ & & \ddots & \\ 0 & & & (V_n^{(n)})^{-1} \end{pmatrix}.$$

For the preconditioners $B_{n,m}^{-1}$ and $C_{n,m}^{-1}$, the inverse of $A_{n/2,m}$ can be represented by the formula in (3.2). This formula can be obtained by solving the following two linear systems:

$$A_{n/2,m}U^{(n/2)} = E^{(n/2)} \quad \text{and} \quad A_{n/2,m}V^{(n/2)} = F^{(n/2)}.$$

These two systems can be solved efficiently by using the PCG method with $B_{n/2,m}$ or $C_{n/2,m}$ as preconditioners. The inverse of $A_{n/4,m}$ involved in the preconditioners $B_{n/2,m}$ and $C_{n/2,m}$ can be recursively generated by using (3.2) until the size of the linear system is sufficiently small. The procedures of recursive computation of $B_{n,m}$ and $C_{n,m}$ are described as follows:

```
Procedure    Input($A_{n,m}$, $n$)  Output($U^{(n)}$, $V^{(n)}$)
    If $k \le N$, then
        solve two linear systems
            $A_{k,m}U^{(k)} = E^{(k)}$   and   $A_{k,m}V^{(k)} = F^{(k)}$
        exactly by direct methods;
    else
        compute $U^{(k/2)}$ and $V^{(k/2)}$ by calling the procedure with the input matrix
        $A_{k/2,m}$ and the integer $k/2$; construct $A_{k/2,m}^{-1}$ by using the output $U^{(k/2)}$
        and $V^{(k/2)}$ via the formula in (3.2);
        solve the two linear systems
            $A_{k,m}U^{(k)} = E^{(k)}$   and   $A_{k,m}V^{(k)} = F^{(k)}$
        by using the PCG method with $B_{k,m}$ (or $C_{k,m}$) as the preconditioner.
```

We remark that if each block of the BT matrix $A_{n,m}$ is Hermitian, then we only need to solve one linear system $A_{n,m}U^{(n)} = E^{(n)}$ in order to represent the inverse of the BT matrix. In this case, the solution $V^{(n)}$ can be obtained by using $U^{(n)}$:

$$V^{(n)} = \begin{pmatrix} U_n^{(n)} \\ U_{n-1}^{(n)} \\ \vdots \\ U_1^{(n)} \end{pmatrix}.$$

**3.1. Computational cost.** The main computational cost of the method comes from the matrix-vector multiplications $A_{n,m}X$, $B_{n,m}^{-1}X$ (or $C_{n,m}^{-1}X$) in each PCG iteration, where $X$ is an $mn$-by-$m$ vector. We note that $A_{n,m}X$ can be computed in $2m$ $2n$-length fast Fourier transforms (FFTs) by first embedding $A_{n,m}$ into a $2mn$-by-$2mn$ block-circulant matrix and then carrying out the multiplication by using the decomposition of the block-circulant matrix. Letting $S_{n,m}$ be the circulant matrix with an $m$-by-$m$ matrix block element, one can find a permutation matrix $P_{n,m}$ such that

$$\overline{S}_{n,m} = (S_{i,j})_{m\times m} = P_{n,m}^{*} W_{n,m} P_{n,m}$$

is a circulant-block matrix, where $S_{i,j}$ is an $n$-by-$n$ circulant matrix. Let $S_{i,j}(:,1)$ denote the first column of the matrix $S_{i,j}$; it is known that $S_{i,j}$ can be diagonalized into an $n\log n$ length FFT, i.e., $S_{i,j} = F^{*}\Lambda_{i,j}F$, where $F$ and $F^{*}$ are the Fourier transform matrix and the inverse Fourier transform matrix, respectively, and $\Lambda_{i,j} = \operatorname{diag}(F\cdot S_{i,j}(:,1))$. Thus we obtain

$$\begin{aligned}\overline{S}_{n,m} &= (I\otimes F^{*})\begin{pmatrix} \Lambda_{11} & \Lambda_{11} & \cdots & \Lambda_{1m} \\ \Lambda_{21} & \Lambda_{22} & \cdots & \Lambda_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \Lambda_{m1} & \Lambda_{m2} & \cdots & \Lambda_{mm} \end{pmatrix}(I\otimes F) \\ &= (I\otimes F^{*})P^{*}DP(I\otimes F),\end{aligned}$$

where $D = \operatorname{diag}(D_1, D_2, \ldots, D_n)$ is a block diagonal matrix, and $[D_k]_{ij} = [\Lambda_{ij}]_{kk}$, i.e., the $(i,j)$th entry of $D_k$ is equal to the $(k,k)$th entry of $\Lambda_{ij}$. Therefore, the block-circulant matrix-vector multiplication can be obtained by

$$S_{n,m}X = P(I\otimes F^{*})P^{*}DP(I\otimes F)P^{*}X.$$

We note that it requires $O(m^2 n\log n)$ operations to compute the block diagonal matrix $D$, and that the block diagonal matrix-vector multiplication requires $O(m^3 n)$ operations. Thus the overall multiplication requires $O(m^2 n\log n + m^3 n)$. For the preconditioner $B_{n,m}$ or $C_{n,m}$, we need to compute matrix-vector products $A_{n/2,m}^{-1}Y$, where $Y$ is an $mn/2$-by-$m$ vector. According to (3.2), the inverse of a BT matrix can be written as the product of lower-triangular BT matrices. Therefore, the matrix-vector multiplication $A_{n/2,m}^{-1}Y$ can be computed by using FFTs by embedding such lower-triangular BT matrices into block-circulant matrices. Such matrix-vector multiplication requires $O(m^2 n\log n + m^3 n)$ operations.

Now we estimate the total cost of recursive computation for solving two linear systems

$$A_{n,m}U^{(n)} = E^{(n)} \quad \text{and} \quad A_{n,m}V^{(n)} = F^{(n)}.$$

For simplicity, we assume $n = 2^{\ell}$. Suppose the number of iterations required for convergence in solving the two $mn_j$-by-$mn_j$ linear systems

$$A_{n_j,m}U^{(n_j)} = E^{(n_j)} \quad \text{and} \quad A_{n_j,m}V^{(n_j)} = F^{(n_j)}, \quad \text{where} \quad n_j = 2^{\nu-j+1},$$

is given by $c_j$ for $j = 1, \ldots, L$. We note that the smallest size of the system is equal to $N = n/2^{\nu-L}$. Therefore the total cost of the recursive computations of $B_{n,m}$ (or $C_{n,m}$) is about $\sum_{j=1}^{L} c_j f_j$, where $f_j$ denotes the cost of each PCG iteration where the size of the system is $n_j$. Since the cost of an $n_j$-length FFT is roughly twice the cost of an $n_j m/2$-length FFT, and the cost of each PCG iteration is $O(m^2 n_j \log n_j + m^3 n_j)$ operations, hence the total cost of the recursive computation is roughly bounded by $O(\max_j\{c_j(m^2 n \log n + m^3 n)\})$.

Next, we compute the operations required for the circulant preconditioners. For the block-circulant matrix $S_{n,m}$, the solution of $S_{n,m}Z = B$ can be obtained by

$$Z = S_{n,m}^{-1}B = P(I \otimes F^*)P^* D^{-1} P(I \otimes F)P^* B.$$

In order to compute the inverse of $D$, $O(m^2 n \log n + m^3 n)$ operations are required. Moreover, the matrix-vector multiplication requires $O(m^3 n)$ operations, and thus $S_{n,m}^{-1}B$ can be computed in $O(m^2 n \log n + m^3 n)$ operations, which is the same complexity of our proposed method. In the next section, we show that our proposed method is competitive with circulant preconditioners.

## 4. Numerical results.

In this section, we test our proposed method. The initial guess is the zero vector. The stopping criterion is

$$\|r_q\|_2 / \|r_0\|_2 \leq 1 \times 10^{-7},$$

where $r_q$ is the residual vector at the $q$th iteration of the PCG method. We use MATLAB 6.1 to conduct the numerical tests. We remark that our preconditioners are constructed recursively. For instance, when we solve $A_{256,m}U^{(256)} = E^{(256)}$, the preconditioners are constructed by solving $A_{128,m}U^{(128)} = E^{(128)}$ and $A_{64,m}U^{(64)} = E^{(64)}$ using the PCG method with the stopping criterion being equal to $10^{-7}$ and using the direct solver for $A_{32,m}U^{(32)} = E^{(32)}$. In all of the tests, the coarsest level is set to be $n = 32$.

In the first test, we consider the following example of a generating function [28]:

$$\begin{pmatrix} 20\sin^2(\theta/2) & |\theta|^{5/2} \\ |\theta|^{5/2} & 20\sin^2(\theta/2) \end{pmatrix}.$$

Table 4.1 shows the corresponding numbers of iterations required for the convergence using our proposed preconditioners $B$ and $C$. As a comparison, the number of iterations from using the preconditioner $M$ studied in [28] is also listed. Our proposed preconditioners are competitive with the preconditioner studied in [28]. We also remark that the construction of our proposed preconditioners does not require the knowledge of the underlying matrix generating function of BT matrices.

TABLE 4.1

*Number of iterations required for convergence.*

| $n$ | $B$ | $C$ | $M$ |
|---|---|---|---|
| 128 | 9 | 4 | 10 |
| 256 | 9 | 5 | 10 |
| 512 | 10 | 5 | 10 |

In the second test, we consider the following four examples.

[illegible] 1.

$$F_3(\theta) = \begin{pmatrix} 2\theta^4 + 1 & |\theta|^3 & \theta^4 \\ |\theta|^3 & 3\theta^4 + 1 & |\theta| \\ \theta^4 & |\theta| & 2\theta^4 + 1 \end{pmatrix}.$$

TABLE 4.2
*Number of iterations required for convergence in Example* 1.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 111 | 13 | 6 | 13 | 12 | 12 | 12 | 12 |
| 128 | 124 | 12 | 6 | 13 | 12 | 12 | 13 | 13 |
| 256 | 133 | 9 | 4 | 13 | 12 | 13 | 13 | 13 |
| 512 | 135 | 8 | 4 | 13 | 13 | 12 | 13 | 12 |
| 1024 | 138 | 5 | 2 | 13 | 13 | 13 | 13 | 13 |
| 2048 | 139 | 2 | 1 | 13 | 13 | 13 | 13 | 13 |
| 4096 | 140 | 2 | 1 | 13 | 13 | 13 | 13 | 13 |

TABLE 4.3
*Number of iterations required for convergence in Example* 2.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 114 | 12 | 6 | 11 | 11 | 11 | 12 | 12 |
| 128 | 172 | 12 | 6 | 12 | 12 | 11 | 12 | 12 |
| 256 | 256 | 12 | 6 | 12 | 13 | 11 | 12 | 13 |
| 512 | 371 | 13 | 6 | 12 | 13 | 12 | 13 | 13 |
| 1024 | 526 | 13 | 6 | 12 | 14 | 12 | 13 | 14 |
| 2048 | 740 | 13 | 6 | 12 | 15 | 12 | 13 | 14 |
| 4096 | $> 1000$ | 13 | 7 | 12 | 15 | 12 | 12 | 14 |

TABLE 4.4
*Number of iterations required for convergence in Example* 3.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 165 | 9 | 4 | 10 | 23 | 13 | 14 | 16 |
| 128 | 354 | 9 | 5 | 10 | 30 | 12 | 13 | 15 |
| 256 | 742 | 10 | 5 | 11 | 40 | 12 | 12 | 13 |
| 512 | $> 1000$ | 10 | 5 | 11 | 54 | 12 | 12 | 13 |
| 1024 | $> 1000$ | 10 | 5 | 11 | $> 1000$ | 12 | 12 | 13 |
| 2048 | $> 1000$ | 10 | 5 | 11 | $> 1000$ | 12 | 12 | 13 |
| 4096 | $> 1000$ | 10 | 6 | 11 | $> 1000$ | 13 | 13 | 15 |

2.

$$F_3(\theta) = \begin{pmatrix} \theta^4 + 1 & |\theta|^3 & |\theta| \\ |\theta|^3 & 2\theta^4 + 1 & \theta^2 \\ |\theta| & \theta^2 & 5|\theta| \end{pmatrix}.$$

3.

$$F_2(\theta) = \begin{pmatrix} 8\theta^2 & (\sin\theta)^4 \\ (\sin\theta)^4 & 8\theta^2 \end{pmatrix}.$$

4.

$$F_3(\theta) = \begin{pmatrix} |\theta| & (\sin\theta)^4 & 0 \\ (\sin\theta)^4 & \theta^2 & (\sin\theta)^8 \\ 0 & (\sin\theta)^8 & \theta^4 \end{pmatrix}.$$

These generating functions are Hermitian matrix-valued functions. Also the generated BT matrices are positive definite. In Example 1, the generated BT matrices are well-conditioned. For Examples 2–4, the generating functions are singular at some points and therefore the corresponding BT matrices are ill-conditioned.

Table 4.5
*Number of iterations required for convergence in Example* 4.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 585 | 19 | 9 | 46 | $> 1000$ | 22 | $> 1000$ | 27 |
| 128 | $> 1000$ | 20 | 10 | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ |
| 256 | $> 1000$ | 24 | 11 | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ |
| 512 | $> 1000$ | 30 | 13 | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ |
| 1024 | $> 1000$ | 36 | 16 | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ |
| 2048 | $> 1000$ | 39 | 25 | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ |
| 4096 | $> 1000$ | 43 | 23 | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ | $> 1000$ |

In Tables 4.2–4.5, we give the number of iterations required for convergence by using $B_{n,m}$ and $C_{n,m}$ as the preconditioners. Here we set the maximum number of iterations to be 1000. If the method does not converge within 1000 iterations, we specify "$> 1000$" in the tables. According to Tables 4.2–4.5, we see that the number of iterations for the nonpreconditioned systems (the column "$I$") increases when the size $n$ increases. However, the number of iterations for the preconditioned systems (the columns "$B$" and "$C$") decreases or almost remains constant when the size $n$ increases in Examples 1–3. The performance of Schur complement preconditioner $C$ is generally better than that of block diagonal preconditioner $B$. We also compare our preconditioners with block-circulant preconditioners; the columns "S" and "T" are the number of iterations required for the Strang and the T. Chan block-circulant preconditioners, respectively. We note that the Strang block-circulant preconditioner may not be positive definite for the ill-conditioned matrix. Indeed, there are several negative eigenvalues of the Strang block-circulant preconditioners in Examples 3 and 4. Even when the Strang circulant preconditioned system converges, the solution may not be correct. We also see from Tables 4.4 and 4.5 that the T. Chan block-circulant preconditioner does not work.

Chan, Ng, and Yip [8, 9] have constructed "best" circulant preconditioners by approximating the generating function with the convolution product that matches the zeros of the generating function. They showed that these circulant preconditioners are effective for ill-conditioned Toeplitz matrices. Here we also construct such "best" block-circulant preconditioners (in the column "$K_i$" and $i$ refers to the order of the kernel that we used) and test their performance. We note from Tables 4.2–4.5 that our proposed preconditioners perform quite well. For Example 4, the method with "best" block-circulant preconditioners does not converge within 1000 iterations.

We remark that for the ill-conditioned systems, a small residual does not necessarily imply an accurate solution. For instance, the systems in Example 4 are very ill-conditioned. We check the accuracy of the solution[1] computed by using the proposed preconditioners and find that the relative errors increase from $10^{-11}$ ($n = 64$) to $10^{-4}$ ($n = 4096$). However, we reiterate that even the other preconditioners do not work.

Also we report the computational times required for convergence in Examples 1–4 in Tables 4.6–4.9, respectively. If the number of iterations is more than 1000, we specify "$**$" in the tables. We see that the computational times required by the block diagonal preconditioner and the Schur complement preconditioner are less than those of the block-circulant preconditioners, especially when $n$ is large. We also note from the tables that the performance of the Schur complement preconditioner is better

---

[1] We set the known solution and compute the corresponding right-hand side for the computation.

Table 4.6
*Computational times required for convergence in Example* 1.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 0.51 | 0.28 | 0.38 | 0.21 | 0.20 | 0.20 | 0.20 | 0.20 |
| 128 | 0.70 | 0.50 | 0.38 | 0.28 | 0.26 | 0.26 | 0.28 | 0.28 |
| 256 | 1.13 | 0.78 | 0.49 | 0.59 | 0.54 | 0.59 | 0.59 | 0.59 |
| 512 | 2.15 | 1.31 | 0.99 | 1.21 | 1.21 | 1.11 | 1.21 | 1.11 |
| 1024 | 4.72 | 2.19 | 0.99 | 3.05 | 3.05 | 3.05 | 3.05 | 3.05 |
| 2048 | 11.63 | 4.79 | 1.18 | 10.94 | 10.94 | 10.94 | 10.94 | 10.94 |
| 4096 | 29.04 | 5.23 | 3.21 | 27.37 | 27.37 | 27.37 | 27.37 | 27.37 |

Table 4.7
*Computational times required for convergence in Example* 2.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 0.52 | 0.26 | 0.21 | 0.18 | 0.18 | 0.18 | 0.20 | 0.20 |
| 128 | 0.97 | 0.50 | 0.38 | 0.26 | 0.26 | 0.24 | 0.26 | 0.26 |
| 256 | 2.18 | 1.04 | 0.73 | 0.54 | 0.59 | 0.50 | 0.54 | 0.60 |
| 512 | 5.92 | 2.13 | 1.49 | 1.11 | 1.21 | 1.11 | 1.20 | 1.21 |
| 1024 | 17.98 | 5.70 | 2.94 | 2.82 | 3.29 | 2.82 | 3.05 | 3.29 |
| 2048 | 61.94 | 11.64 | 7.08 | 10.10 | 12.63 | 10.10 | 10.94 | 11.78 |
| 4096 | ** | 34.02 | 22.50 | 25.27 | 31.58 | 25.27 | 25.27 | 29.48 |

Table 4.8
*Computational times required for convergence in Example* 3.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 0.75 | 0.20 | 0.14 | 0.16 | 0.38 | 0.21 | 0.23 | 0.26 |
| 128 | 1.99 | 0.38 | 0.32 | 0.22 | 0.65 | 0.26 | 0.28 | 0.33 |
| 256 | 6.04 | 0.87 | 0.61 | 0.50 | 1.81 | 0.54 | 0.54 | 0.59 |
| 512 | ** | 1.64 | 1.24 | 1.02 | 5.01 | 1.11 | 1.11 | 1.21 |
| 1024 | ** | 4.39 | 2.45 | 2.58 | ** | 2.82 | 2.82 | 3.05 |
| 2048 | ** | 8.95 | 5.90 | 9.25 | ** | 10.10 | 10.10 | 10.94 |
| 4096 | ** | 26.17 | 19.29 | 23.16 | ** | 27.37 | 27.37 | 31.58 |

Table 4.9
*Computational times required for convergence in Example* 4.

| $n$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| 64 | 2.67 | 0.41 | 0.31 | 0.75 | ** | 0.36 | ** | 0.44 |
| 128 | ** | 0.83 | 0.64 | ** | ** | ** | ** | ** |
| 256 | ** | 2.08 | 1.34 | ** | ** | ** | ** | ** |
| 512 | ** | 4.92 | 3.22 | ** | ** | ** | ** | ** |
| 1024 | ** | 15.79 | 7.83 | ** | ** | ** | ** | ** |
| 2048 | ** | 34.91 | 29.50 | ** | ** | ** | ** | ** |
| 4096 | ** | 112.53 | 73.93 | ** | ** | ** | ** | ** |

than that of the block diagonal preconditioner.

To illustrate the fast convergence of the proposed method, in Table 4.10, we calculate the number of eigenvalues within the small interval for $n = 128$ in Examples 1–4. We find that the spectra of the preconditioned matrices $C_{n,m}^{-1}A_{n,m}$ and $B_{n,m}^{-1}A_{n,m}$ are closer to 1 than those of circulant preconditioners and no preconditioner.

Finally, we report that the numbers of iterations are about the same even when the stopping criteria $\tau$ of the PCG method at each level in the recursive calculation of the proposed preconditioners is $1 \times 10^{-3}$, $1 \times 10^{-4}$, and $1 \times 10^{-7}$ for the proposed preconditioners.

Next, we consider an application of our algorithm to BT systems arising from

Table 4.10
*The percentages of the number of eigenvalues within the interval of* $[0.99, 1.01]$ *for* $n = 128$.

| | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|
| Example 1 | 2.60% | 94.27% | 98.44% | 77.34% | 30.99% | 93.23% | 87.76% | 77.08% |
| Example 2 | 4.43% | 94.79% | 98.44% | 84.90% | 46.62% | 94.27% | 94.01% | 85.16% |
| Example 3 | 0.00% | 95.31% | 99.22% | 89.84% | 56.64% | 84.38% | 81.64% | 78.91% |
| Example 4 | 0.52% | 93.23% | 98.18% | 81.77% | 51.30% | 79.95% | 73.44% | 70.05% |

multichannel least squares filtering. Another application to queueing networks can be found in the full report at ftp://ftp.math.hkbu.edu.hk/pub/techreport/math431.pdf.

**Application I:** Multichannel least squares filtering is a data processing method that makes use of the signals from each of the $m$ channels. We represent this multichannel data by $\mathbf{x}_t$, where $\mathbf{x}_t$ is a column vector whose elements are the signals from each channel. Since we are interested in digital processing methods, we suppose that the signals are sampled at discrete, equally spaced time points which are represented by the time index $t$. Without loss of generality, we require that $t$ take on successive integer values. If we let $x_{it}$ represents the signal coming from the $i$th channel $(i = 1, 2, \ldots, m)$, the multichannel signal can be written as $\mathbf{x}_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{m,t} \end{pmatrix}$. The filter is represented by the coefficients $S_1, S_2, \ldots, S_n$, where each coefficient $S_k$ $(k = 1, 2, \ldots, n)$ is an $n$-by-$m$ matrix. The multichannel signal $\mathbf{x}_t$ received by the array system represents the input to the filter and the resulting output of the filter is a multichannel signal, which we denote by the column vector $\mathbf{y}_t = \begin{pmatrix} y_{1,t} \\ y_{2,t} \\ \vdots \\ y_{m,t} \end{pmatrix}$. The relationship between input $\mathbf{x}_t$ and output $\mathbf{y}_t$ is given by the convolution formula $\mathbf{y}_t = S_1\mathbf{x}_t + S_2\mathbf{x}_{t-1} + \cdots + S_n\mathbf{x}_{t-n+1}$. The determination of the filter coefficients is based on the concept of a desired output denoted by a column vector $\mathbf{z}_t = \begin{pmatrix} z_{1,t} \\ z_{2,t} \\ \vdots \\ z_{m,t} \end{pmatrix}$. On each channel $(i = 1, 2, \ldots, m)$, there will be an error between the desired output $\mathbf{z}_t$ and the actual output $\mathbf{y}_t$. The mean square value of this error is given by $\mathcal{E}[(\mathbf{z}_t - \mathbf{y}_t)^2]$. The sum of the mean square errors for all the channels is $\sum_{i=1}^{m} \mathcal{E}[(\mathbf{z}_t - \mathbf{y}_t)^2]$. The least squares determination of the filter coefficients requires that this sum be minimum. This minimization leads to a set of linear equations

$$\begin{pmatrix} R_0 & R_1 & \cdots & R_{n-1} \\ R_1 & R_0 & \cdots & R_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ R_{n-1} & R_{n-2} & \cdots & R_0 \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_n \end{pmatrix} = \begin{pmatrix} G_1 \\ G_2 \\ \vdots \\ G_n \end{pmatrix}, \tag{4.1}$$

where

$$R_j = \mathcal{E}[\mathbf{x}_t\mathbf{x}_{t-j}^*] \quad \text{and} \quad G_j = \mathcal{E}[\mathbf{z}_t\mathbf{x}_{t-j+1}^*].$$

Here $R_j$ is an $m$-by-$m$ matrix and is the autocorrelation coefficients of the input signal $\mathbf{x}_t$, and $G_j$ is an $n$-by-$m$ matrix and is the cross-correlation coefficients between the desired output $\mathbf{z}_t$ and the input signal $\mathbf{x}_t$.

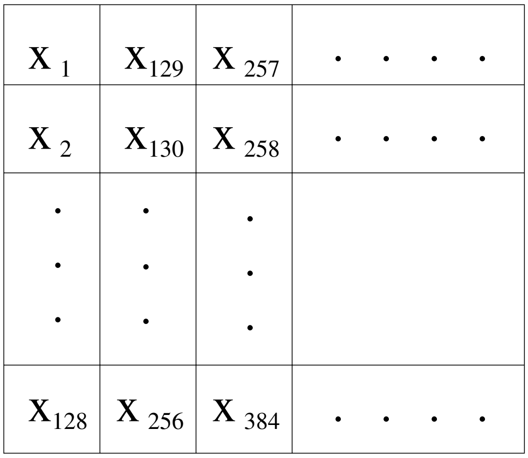


Fig. 4.1. *Color image and data vectors.*

In the test, a 128-by-128 color image is used to generate the data points. We consider the pixel value of the color image to be $\mathbf{x}_t$ $(t = 1, 2, \ldots, 128^2)$; see Figure 4.1. We remark that color can be regarded as a set of three images in their primary color components: red, green, and blue. In the least squares filtering, there are three channels, i.e., $m = 3$. Our task is to generate the multichannel least squares filters such that the sum of the mean square errors for all the channels

$$\sum_{i=1}^{m} \mathcal{E}\{\mathbf{x}_{t+1} - [S_1\mathbf{x}_t + S_2\mathbf{x}_{t-1} + \cdots + S_n\mathbf{x}_{t-n+1}]^2\}$$

is minimum. Such least squares filters have been commonly used in color image processing for coding and enhancement [20]. Table 4.11 shows the number of iterations required for convergence. Table 4.12 shows the number of iterations required for convergence when more synthetic multichannel data sets are generated to test. The stopping criteria are the same as those for Tables 4.2–4.5. Notice that the generating function of the BT matrices are unknown in this case. However, the construction of the proposed preconditioners only requires the entries of $A_{n,m}$ and does not require the explicit knowledge of the generating function $F_m(\theta)$ of $A_{n,m}$. We find that the generated BT matrices are very ill-conditioned. Therefore, the number of iterations required for convergence without preconditioning is very large, but the performance of the preconditioners $B_{n,m}$ and $C_{n,m}$ is very good. We also check the accuracy of the solution[2] computed by using the proposed preconditioners and find that the relative errors are about $10^{-9}$. These results show that our proposed preconditioner performs quite well.

We also generate more synthetic multichannel data sets to test the performance of our proposed method for larger $m$. Table 4.12 shows the number of iterations required for convergence. The stopping criteria are the same as those for Tables 4.2–4.5. The results show that our proposed preconditioner performs quite well.

---

[2] We set the known solution and compute the corresponding right-hand side for the computation.

Table 4.11
*Number of iterations required for convergence.*

| $n$ | $m$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|---|
| 16 | 3 | 166 | 18 | 9 | 168 | 22 | 36 | 39 | 50 |
| 32 | 3 | 725 | 26 | 13 | >1000 | 21 | 32 | 36 | 43 |
| 64 | 3 | 725 | 26 | 13 | >1000 | 15 | 29 | 31 | 37 |
| 128 | 3 | >1000 | 60 | 30 | >1000 | 42 | >1000 | >1000 | >1000 |
| 256 | 3 | >1000 | 85 | 40 | > 1000 | > 1000 | >1000 | >1000 | >1000 |
| 512 | 3 | > 1000 | 95 | 44 | > 1000 | > 1000 | >1000 | >1000 | >1000 |
| 1024 | 3 | > 1000 | 101 | 51 | > 1000 | > 1000 | >1000 | >1000 | >1000 |

Table 4.12
*Number of iterations required for convergence.*

| $n$ | $m$ | $I$ | $B$ | $C$ | S | T | $K_4$ | $K_6$ | $K_8$ |
|---|---|---|---|---|---|---|---|---|---|
| 16 | 6 | 473 | 26 | 13 | 503 | 23 | 45 | 52 | 66 |
| 32 | 6 | 958 | 30 | 15 | > 1000 | 28 | 42 | 43 | 56 |
| 64 | 6 | > 1000 | 39 | 18 | > 1000 | 28 | 39 | 41 | 50 |
| 128 | 6 | > 1000 | 50 | 25 | > 1000 | 44 | > 1000 | > 1000 | > 1000 |
| 16 | 9 | 731 | 38 | 18 | 945 | 31 | 58 | 62 | 81 |
| 32 | 9 | > 1000 | 42 | 21 | > 1000 | 35 | 55 | 58 | 72 |
| 64 | 9 | > 1000 | 53 | 25 | > 1000 | 35 | 59 | 65 | 75 |
| 128 | 9 | > 1000 | 70 | 35 | > 1000 | 68 | > 1000 | > 1000 | > 1000 |
| 16 | 12 | 989 | 44 | 22 | > 1000 | 36 | 65 | 71 | 94 |
| 32 | 12 | > 1000 | 50 | 25 | > 1000 | 40 | 63 | 66 | 87 |
| 64 | 12 | > 1000 | 64 | 31 | > 1000 | 42 | 75 | 81 | > 1000 |
| 128 | 12 | > 1000 | 103 | 47 | > 1000 | > 1000 | > 1000 | > 1000 | > 1000 |

**5. Concluding remarks.** In this paper, we proposed block diagonal and Schur complement preconditioners for BT matrices. We have proved that for some BT coefficient matrices, the spectra of the preconditioned matrices are uniformly bounded except for a fixed number of outliers, where the number of outliers depends on $m$. Therefore the conjugate gradient method will converge very quickly when applied to solving the preconditioned systems, especially when $m$ is small. Our experimental results show that the Schur-complement preconditioner is always better than the block diagonal preconditioner. Applications to BT systems arising from least squares filtering problems and queueing networks were discussed. The method can also be applied to solve other nonsymmetric problems that arise in other queueing systems [11].

**Acknowledgments.** The authors are very much indebted to Prof. Per Christian Hansen and the referees for their valuable comments and suggestions which greatly improved this paper.

## REFERENCES

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.

[2] A. Ben-Artzi and T. Shalom, *On inversion of Toeplitz and close to Toeplitz matrices*, Linear Algebra Appl., 75 (1986), pp. 173–192.

[3] F. Di Benedetto and S. Serra Capizzano, *A unifying approach to abstract matrix algebra preconditioning*, Numer. Math., 82 (1999), pp. 57–90.

[4] R. Chan, *Circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 542–550.

[5] R. Chan, *Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions*, IMA J. Numer. Anal., 11 (1991), pp. 333–345.

[6] R. Chan, Q. Chang, and H.-N. Sun, *Multigrid method for ill-conditioned symmetric Toeplitz systems*, SIAM J. Sci. Comput., 19 (1998), pp. 516–529.

[7] R. Chan and M. Ng, *Conjugate gradient methods for Toeplitz systems*, SIAM Rev., 38 (1996), pp. 427–482.

[8] R. Chan, M. Ng, and A. Yip, *The best circulant preconditioners for Hermitian Toeplitz systems* II: *The multiple-zero case*, Numer. Math., 92 (2002), pp. 17–40.

[9] R. Chan, A. Yip, and M. Ng, *The best circulant preconditioners for Hermitian Toeplitz systems*, SIAM J. Numer. Anal., 38 (2001), pp. 876–896.

[10] T. Chan, *An optimal circulant preconditioner for Toeplitz systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 766–771.

[11] W. Ching, *Iterative Methods for Queuing and Manufacturing Systems*, Springer Monographs in Mathematics 1, Springer-Verlag, London, 2001.

[12] G. Fiorentino and S. Serra, *Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions*, SIAM J. Sci. Comput., 17 (1996), pp. 1068–1081.

[13] I. Gohberg and A. Semencul, *The inversion of finite Toeplitz matrices and their continuous analogs*, Mat. Issled., 7 (1972), pp. 201–223.

[14] I. Gohberg and G. Heinig, *Inversion of finite-section Toeplitz matrices consisting of elements of noncommutative algebra*, Rev. Roumaine Math. Pures Appl., 19 (1974), pp. 623–663.

[15] G. Heinig, *On the reconstruction of Toeplitz matrix inverses from columns*, Linear Algebra Appl., 350 (2002), pp. 199–212.

[16] G. Heinig and L. Rost, *Algebraic Methods for Toeplitz-Like Matrices and Operators*, Birkhäuser Verlag, Basel, 1984.

[17] T. Huckle, *Circulant and skewcirculant matrices for solving Toeplitz matrix problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 767–777.

[18] X. Jin, *Band Toeplitz preconditioners for block Toeplitz systems*, J. Comput. Appl. Math., 70 (1996), pp. 225–230.

[19] G. Labahn and T. Shalom, *Inversion of Toeplitz matrices with only two standard equations*, Linear Algebra Appl., 175 (1992), pp. 143–158.

[20] J. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[21] M. Miranda and P. Tilli, *Asymptotic spectra of Hermitian block Toeplitz matrices and preconditioning results*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 867–881.

[22] M.K. Ng, *Band preconditioners for block-Toeplitz–Toeplitz-block systems*, Linear Algebra Appl., 259 (1997), pp. 307–327.

[23] M.K. Ng, K. Rost, and Y.-W. Wen, *On inversion of Toeplitz matrices*, Linear Algebra Appl., 348 (2002), pp. 145–151.

[24] M.K. Ng, H. Sun, and X. Jin, *Recursive-based PCG methods for Toeplitz systems with nonnegative generating functions*, SIAM J. Sci. Comput., 24 (2003), pp. 1507–1529.

[25] D. Potts and G. Steidl, *Preconditioners for ill-conditioned Toeplitz systems constructed from positive kernels*, SIAM J. Sci. Comput., 22 (2000), pp. 1741–1761.

[26] M. Priestley, *Spectral Analysis and Time Series*, Vol. 1, Academic Press, London–New York, 1981.

[27] S. Serra, *Preconditioning strategies for asymptotically ill-conditioned block Toeplitz systems*, BIT, 34 (1994), pp. 579–594.

[28] S. Serra, *Spectral and computational analysis of block Toeplitz matrices having nonnegative definite matrix-valued generating functions*, BIT, 39 (1999), pp. 152–175.

[29] S. Serra, *Asymptotic results on the spectra of block Toeplitz preconditioned matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 31–44.

[30] G. Strang, *A proposal for Toeplitz matrix calculations*, Stud. Appl. Math., 74 (1986), pp. 171–176.

[31] H.-W. Sun, X. Jin, and Q. Chang, *Convergence of the multigrid method for ill-conditioned block Toeplitz systems*, BIT, 41 (2001), pp. 179–190.

[32] W. Trench, *An algorithm for the inversion of finite Toeplitz matrices*, J. Soc. Indust. Appl. Math., 12 (1964), pp. 515–522.

[33] E. Tyrtyshnikov, *Optimal and superoptimal circulant preconditioners*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 459–473.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1120–1146

# MATRIX NEARNESS PROBLEMS WITH BREGMAN DIVERGENCES*

INDERJIT S. DHILLON† AND JOEL A. TROPP‡

**Abstract.** This paper discusses a new class of matrix nearness problems that measure approximation error using a directed distance measure called a *Bregman divergence*. Bregman divergences offer an important generalization of the squared Frobenius norm and relative entropy, and they all share fundamental geometric properties. In addition, these divergences are intimately connected with exponential families of probability distributions. Therefore, it is natural to study matrix approximation problems with respect to Bregman divergences. This article proposes a framework for studying these problems, discusses some specific matrix nearness problems, and provides algorithms for solving them numerically. These algorithms apply to many classical and novel problems, and they admit a striking geometric interpretation.

**Key words.** matrix nearness problems, Bregman divergences, squared Euclidean distance, relative entropy, alternating projections

**AMS subject classifications.** 15A99, 65F30, 90C25

**DOI.** 10.1137/060649021

## 1. Introduction.

A recurring problem in matrix theory is to find a structured matrix that best approximates a given matrix with respect to some distance measure. For example, it may be known a priori that a certain constraint ought to hold, and yet it fails on account of measurement errors or numerical roundoff. An attractive remedy is to replace the tainted matrix by the nearest matrix that does satisfy the constraint. Matrix approximation problems typically measure the distance between matrices with a norm. The Frobenius and spectral norms are pervasive choices because they are so analytically tractable. Nevertheless, these norms are not always defensible in applications, where it may be wiser to tailor the distance measure to the context.

In this paper, we discuss a new class of matrix nearness problems that use a directed distance measure called a [illegible]. Given a differentiable, strictly convex function $\varphi$ that maps matrices to the extended real numbers, we define the Bregman divergence of the matrix $\boldsymbol{X}$ from the matrix $\boldsymbol{Y}$ as

$$D_\varphi(\boldsymbol{X};\boldsymbol{Y}) \overset{\text{def}}{=} \varphi(\boldsymbol{X}) - \varphi(\boldsymbol{Y}) - \langle \nabla\varphi(\boldsymbol{Y}), \boldsymbol{X}-\boldsymbol{Y}\rangle\,,$$

where the inner product $\langle \boldsymbol{X},\boldsymbol{Y}\rangle = \operatorname{Re}\operatorname{Tr}\boldsymbol{X}\boldsymbol{Y}^*$. The two principal examples of Bregman divergences deserve immediate mention. When $\varphi(\boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{X}\|_F^2$, the associated divergence is the squared Frobenius norm $\frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2$. When $\varphi$ is the negative Shannon entropy, we obtain the Kullback–Leibler divergence, which is also known as relative entropy. But these two cases are just the tip of the iceberg.

Bregman divergences are well suited for nearness problems because they share many geometric properties with the squared Frobenius norm. They also exhibit an

---

*Received by the editors January 4, 2006; accepted for publication (in revised form) by N. J. Higham March 20, 2007; published electronically November 21, 2007.
http://www.siam.org/journals/simax/29-4/64902.html

†Department of Computer Sciences, University of Texas, Austin, TX 78712-1188 (inderjit@cs.utexas.edu). This author's research was supported by NSF grant CCF-0431257, NSF career award ACI-0093404, and NSF-ITR award IIS-0325116.

‡Applied and Computational Mathematics, California Institute of Technology, Pasadena, CA 91125-5000 (jtropp@acm.caltech.edu). This author's research was supported by an NSF graduate fellowship.

intimate relationship with exponential families of probability distributions, which recommends them for solving problems that arise in the statistical analysis of data. We will elaborate on these connections in what follows.

Let us begin with a formal statement of the Bregman nearness problem. Suppose that $D_\varphi$ is a Bregman divergence, and suppose that $\{C_k\}$ is a finite collection of closed, convex sets whose intersection is nonempty. Given an input matrix $\boldsymbol{Y}$, our goal is to produce a matrix $\boldsymbol{X}$ in the intersection that diverges the least from $\boldsymbol{Y}$, i.e., to solve

$$\min_{\boldsymbol{X}} \; D_\varphi(\boldsymbol{X}; \boldsymbol{Y}) \qquad \text{subject to} \qquad \boldsymbol{X} \in \bigcap\nolimits_k C_k. \tag{1.1}$$

Under mild conditions, the solution to (1.1) is unique, and it has a variational characterization analogous with the characterization of an orthogonal projection onto a convex set [10]. Minimization with respect to the second argument of the divergence enjoys rather less structure, so we refer the reader to [5] for more details. A major advantage of our problem formulation is that it admits a natural algorithm. If one possesses a method for minimizing the divergence over each of the constraint sets, then it is possible to solve (1.1) by minimizing over each constraint in turn while introducing a series of simple corrections. Several classical algorithms from the matrix literature fit into this geometric framework, but it also provides an approach to many novel problems.

We view this paper as an expository work with two central goals. First, it introduces Bregman divergences to the matrix theory literature, and it argues that they provide an important and natural class of distance measures for matrix nearness problems. Moreover, the article unifies a large class of problems into a geometrical framework, and it shows that these problems can be solved with a set of classical algorithms. Second, the paper provides specific examples of nearness problems with respect to Bregman divergences. One example is the familiar problem of producing the nearest contingency table with fixed marginals. Novel examples include computing matrix approximations using the minimum Bregman information (MBI) principle, identifying the metric graph nearest to an arbitrary graph, and determining the nearest correlation and kernel matrix with respect to matrix divergences, such as the von Neumann divergence. These applications show how Bregman divergences can be used to preserve and exploit additional structure that appears in a problem.

We must warn the reader that, in spite of the availability of some general purpose algorithms for working with Bregman divergences, they may require a substantial amount of computational effort. One basic reason is that nearness problems with respect to the Frobenius norm usually remain within the domain of linear algebra, which is a developed technology. Bregman divergences, on the other hand, transport us to the world of convex optimization, which is a rougher frontier. As outlined in section 8, there remain many unresolved research issues on the computational aspects of Bregman divergences.

Here is a brief outline of the article. Section 2 introduces Bregman divergences and Bregman projections along with their connection to exponential families of probability distributions. Matrix Bregman divergences that depend on the spectral properties of a matrix are covered in subsection 2.6. Section 3 discusses numerical methods for the basic problem of minimizing a Bregman divergence over a hyperplane. In section 4, we develop the successive projection algorithm for solving the Bregman nearness problem subject to affine constraints. Section 5 gives several examples of these problems: finding the nearest contingency table with fixed marginals, computing matrix approximations for data analysis, and determining the nearest correlation matrix with respect to the von Neumann divergence. Section 6 presents the successive projection–correction algorithm for solving the Bregman nearness problem subject to a polyhedral constraint. In section 7 we discuss two matrix nearness problems with nonaffine constraints: finding the nearest metric graph and learning a kernel matrix for data mining and machine learning applications.

## 2. Bregman divergences and Bregman projections.

This section develops the directed distance functions that were first studied by Bregman [8]. Our primary source is the superb article of Bauschke and Borwein [4], which studies a subclass of Bregman divergences that exhibits many desirable properties in connection with nearness problems like (1.1).

### 2.1. Convex analysis.

The literature on Bregman divergences involves a significant amount of convex analysis. Some standard references for this material are [35, 20]. We review some of these ideas in an effort to make this article accessible to readers who are less familiar with this field.

We will work in a finite-dimensional, real inner-product space $\mathscr{X}$. The real-linear inner product is denoted by $\langle\cdot,\cdot\rangle$ and the induced norm by $\|\cdot\|_2$. In general, the elements of $\mathscr{X}$ will be expressed with lowercase bold italic letters such as $\boldsymbol{x}$ and $\boldsymbol{y}$. We will switch to capitals, such as $\boldsymbol{X}$ and $\boldsymbol{Y}$, when it is important to view the elements of $\mathscr{X}$ as matrices.

A [illegible] is a subset $C$ of $\mathscr{X}$ that exhibits the property

$$s\,\boldsymbol{x} + (1-s)\,\boldsymbol{y} \in C \qquad \text{for all } s \in (0,1) \text{ and } \boldsymbol{x}, \boldsymbol{y} \in C.$$

In words, the line segment connecting each pair of points in a convex set falls within the set. The [illegible] of a convex set, abbreviated ri, is the interior of that set considered as a subset of the lowest-dimensional affine space that contains it.

In convex analysis, functions are defined on all of $\mathscr{X}$, and they take values in the extended real numbers, $\mathbb{R} \cup \{\pm\infty\}$. The [illegible] of a function $f$ is the set

$$\operatorname{dom} f \overset{\text{def}}{=} \{\boldsymbol{x} \in \mathscr{X} : f(\boldsymbol{x}) < +\infty\}.$$

A function $f$ is [illegible] if its domain is convex and it verifies the inequality

$$f(s\,\boldsymbol{x} + (1-s)\,\boldsymbol{y}) \le s\,f(\boldsymbol{x}) + (1-s)\,f(\boldsymbol{y}) \quad \text{for all } s \in (0,1) \text{ and } \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} f.$$

If the inequality is strict, then $f$ is [illegible]. In words, the chord connecting each pair of points on the graph of a (strictly) convex function lies (strictly) above the graph. A convex function is [illegible] if it takes at least one finite value and never takes the value $-\infty$. A convex function $f$ is [illegible] if its lower level set $\{\boldsymbol{x} : f(\boldsymbol{x}) \le \alpha\}$ is closed for each real $\alpha$. In particular, a convex function is closed whenever its domain is closed (but not conversely).

For completeness, we also introduce some technical definitions that the casual reader may prefer to glide through. A proper convex function $f$ is called [illegible] if it is everywhere differentiable on the (nonempty) interior of its domain and if $\|\nabla f(\boldsymbol{x}_t)\|$ tends to infinity for every sequence $\{\boldsymbol{x}_t\}$ from $\operatorname{ri}(\operatorname{dom} f)$ that converges to a point on the boundary of $\operatorname{dom} f$. Roughly speaking, an essentially smooth function cannot be extended to a convex function with a larger domain. The function $f(x) = -\log(x)$ with domain $(0,+\infty)$ is an example of an essentially smooth function. In what follows, we will focus on convex functions of [illegible]. A Legendre function is a closed, proper, convex function that is essentially smooth and also strictly convex on the relative interior of its domain.

Every convex function has a dual representation in terms of its supporting hyperplanes. This idea is formalized in the *Fenchel conjugate*, which is defined as

$$f^*(\boldsymbol{\theta}) \overset{\text{def}}{=} \sup_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \boldsymbol{\theta} \rangle - f(\boldsymbol{x}) \right\}.$$

No confusion should arise from our usage of the symbol $^*$ for complex-conjugate transposition as well as Fenchel conjugation. The following facts are valuable. The conjugate of a convex function is always closed and convex. If $f$ is a closed, convex function, then $(f^*)^* = f$. A convex function has Legendre type if and only if its conjugate has Legendre type.

Finally, we say that a convex function $f$ is *cofinite* when

$$\lim_{\xi \to \infty} f(\xi\, \boldsymbol{x})/\xi = +\infty \qquad \text{for all nonzero } \boldsymbol{x} \text{ in } \mathscr{X}.$$

This definition means that a cofinite function grows superlinearly in every direction. For example, the function $\|\cdot\|_2^2$ is cofinite, but the function $\exp(\cdot)$ is not. It can be shown that a closed, proper, convex function $f$ is cofinite if and only if $\operatorname{dom} f^* = \mathscr{X}$.

**2.2. Divergences.** Suppose that $\varphi$ is a convex function of Legendre type. From every such seed function, we may construct a Bregman divergence[1]

$$D_\varphi : \operatorname{dom}\varphi \times \operatorname{ri}(\operatorname{dom}\varphi) \to [0, +\infty)$$

via the rule

$$D_\varphi(\boldsymbol{x}; \boldsymbol{y}) \overset{\text{def}}{=} \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y}) - \langle \nabla\varphi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle .$$

Geometrically, the divergence calculates how much the supporting hyperplane to $\varphi$ at $\boldsymbol{y}$ underestimates the value of $\varphi(\boldsymbol{x})$. For an illustration, see Figure 2.1. A Bregman divergence equals zero whenever $\boldsymbol{x} = \boldsymbol{y}$, and it is positive otherwise. It is strictly convex in its first argument, and it is jointly continuous in both arguments.

As a first example, consider the seed function $\varphi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, which is a Legendre function on all of $\mathscr{X}$. The associated divergence is

$$D_\varphi(\boldsymbol{x}; \boldsymbol{y}) = \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 .$$

We will refer to this function as the Euclidean divergence. Observe that it is symmetric in its two arguments, but it does not satisfy the triangle inequality.

Another basic example arises from the negative Shannon entropy,

$$\varphi(\boldsymbol{x}) = \sum\nolimits_n x_n \log x_n - x_n,$$

where we place the convention that $0 \log 0 = 0$. This entropy is a Legendre function on the nonnegative orthant, and it yields the divergence

$$D_\varphi(\boldsymbol{x}; \boldsymbol{y}) = \sum\nolimits_n \left[ x_n \log \frac{x_n}{y_n} - x_n + y_n \right], \tag{2.1}$$

---

[1] It is also possible to define Bregman divergences with respect to any differentiable, strictly convex function. These divergences are not necessarily well behaved.

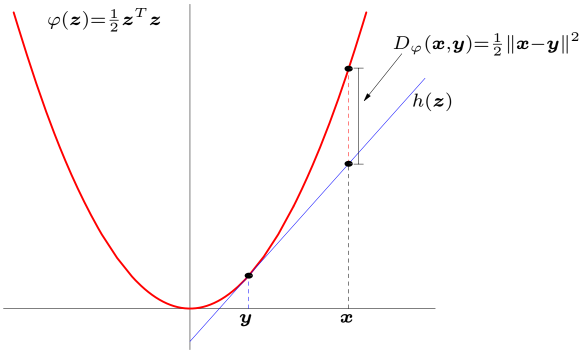

$\varphi(z)=\frac{1}{2}z^Tz$
$D_\varphi(x,y)=\frac{1}{2}\|x-y\|^2$
$h(z)$
$y$
$x$

FIG. 2.1. *An example of a Bregman divergence is the squared Euclidean distance. The Bregman divergence $D_\varphi(\boldsymbol{x};\boldsymbol{y})$ calculates how much the supporting hyperplane to $\varphi$ at $\boldsymbol{y}$ underestimates the value of $\varphi(\boldsymbol{x})$.*

which is variously called the [illegible], the [illegible], or the [illegible]. This divergence is not symmetric, and it does not satisfy the triangle inequality.

Bregman divergences are often referred to as [illegible], but this terminology is misleading. A Bregman divergence should not be viewed as a generalization of a metric but rather as a generalization of the preceding two examples. Like a metric, every Bregman divergence is positive except when its arguments coincide. On the other hand, divergences do not generally satisfy the triangle inequality, and they are symmetric only when the seed function $\varphi$ is quadratic. In compensation, divergences exhibit other structural properties. For every three points in the interior of dom $\varphi$, we have the relation

$$D_\varphi(\boldsymbol{x};\boldsymbol{z}) = D_\varphi(\boldsymbol{x};\boldsymbol{y}) + D_\varphi(\boldsymbol{y};\boldsymbol{z}) - \langle \nabla\varphi(\boldsymbol{z}) - \nabla\varphi(\boldsymbol{y}), \boldsymbol{x}-\boldsymbol{y}\rangle .$$

When $D_\varphi$ is the Euclidean divergence, one may identify this formula as the law of cosines. Later, we will also encounter a Pythagorean theorem.

We also note another expression for the divergence, which emphasizes that it is a sort of locally quadratic distance measure,

$$D_\varphi(\boldsymbol{x};\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^* \left\{\nabla^2\varphi(\boldsymbol{\xi})\right\} (\boldsymbol{x}-\boldsymbol{y}),$$

where $\boldsymbol{\xi}$ is an unknown vector that depends on $\boldsymbol{x}$ and $\boldsymbol{y}$. This formula can be obtained from the Taylor expansion of the seed function with an exact remainder term.

**2.3. Exponential families.** Suppose that $\psi$ is a Legendre function. A [illegible] is a parameterized family of probability distributions on $\mathscr{X}$ with density function (with respect to the Lebesgue measure on $\mathscr{X}$) of the form

$$p_\psi(\boldsymbol{x}\,|\,\boldsymbol{\theta}) = \exp\{\langle \boldsymbol{x},\boldsymbol{\theta}\rangle - \psi(\boldsymbol{\theta}) - h(\boldsymbol{x})\},$$

where the parameter $\boldsymbol{\theta}$ is drawn from the open set dom $\psi$ [3]. The function $\psi$ is called the [illegible] of the exponential family, and it completely determines the function $h$. The [illegible] of the distribution $p_\psi(\,\cdot\,|\,\boldsymbol{\theta})$ is the vector

$$\boldsymbol{\mu}(\boldsymbol{\theta}) \overset{\text{def}}{=} \int_{\mathscr{X}} \boldsymbol{x}\, p_\psi(\boldsymbol{x}\,|\,\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{x},$$

where d$\boldsymbol{x}$ denotes the Lebesgue measure on $\mathscr{X}$. Many common probability distributions belong to exponential families. Prominent examples include Gaussian, Poisson, Bernoulli, and gamma distributions.

It has recently been established that there is a unique Bregman divergence that corresponds to every regular exponential family.

Theorem 1 (Banerjee et al. [2]). [illegible] $\varphi$ [illegible] $\psi$ [illegible] $D_\varphi$ [illegible] $\varphi$ [illegible] $p_\psi(\,\cdot\,|\,\boldsymbol{\theta})$ [illegible] $\psi$ [illegible]

$$p_\psi(\boldsymbol{x}\,|\,\boldsymbol{\theta}) = \exp\{-D_\varphi(\boldsymbol{x};\boldsymbol{\mu}(\boldsymbol{\theta}))\}\, g_\varphi(\boldsymbol{x}),$$

[illegible] $g_\varphi$ [illegible] $\varphi$

The spherical Gaussian distribution provides an especially interesting example of this relationship [2]. Suppose that $\boldsymbol{\mu}$ is an arbitrary vector in $\mathscr{X}$, and let $\sigma^2$ be a fixed positive number. The spherical Gaussian distributions with mean $\boldsymbol{\mu}$ and variance $\sigma^2$ form an exponential family with parameter $\boldsymbol{\theta} = \boldsymbol{\mu}/\sigma^2$ and cumulant function $\psi(\boldsymbol{\theta}) = \frac{\sigma^2}{2}\,\|\boldsymbol{\theta}\|_2^2$. The Fenchel conjugate of the cumulant function is $\varphi(\boldsymbol{x}) = \frac{1}{2\sigma^2}\,\|\boldsymbol{x}\|_2^2$, and so the Bregman divergence that appears in the bijection theorem is

$$D_\varphi(\boldsymbol{x};\boldsymbol{\mu}) = \frac{1}{2\sigma^2}\,\|\boldsymbol{x}-\boldsymbol{\mu}\|_2^2\,.$$

We see that the density of the distribution at a point $\boldsymbol{x}$ depends essentially on the Bregman divergence of $\boldsymbol{x}$ from the mean vector $\boldsymbol{\mu}$. This observation reinforces the intuition that the squared Euclidean norm enjoys a profound relationship with Gaussian random variables.

### 2.4. Bregman projections.

Suppose that $\varphi$ is a convex function of Legendre type, and let $C$ be a closed, convex set that intersects ri(dom $\varphi$). Given a point $\boldsymbol{y}$ from ri(dom $\varphi$), we may pose the minimization problem

$$\min_{\boldsymbol{x}}\; D_\varphi(\boldsymbol{x};\boldsymbol{y}) \qquad \text{subject to} \qquad \boldsymbol{x}\in C\cap \operatorname{ri}(\operatorname{dom}\varphi). \tag{2.2}$$

Since $D_\varphi(\,\cdot\,;\boldsymbol{y})$ is strictly convex, it follows from a standard argument that there exists [illegible] one minimizer. It can be shown that, when $\varphi$ is a Legendre function, there exists [illegible] one minimizer [4, Theorem 3.12]. Therefore, the problem (2.2) has a single solution, which is called the [illegible] of $\boldsymbol{y}$ onto $C$ with respect to the divergence $D_\varphi$. Denote this solution by $P_C(\boldsymbol{y})$, and observe that we have defined a map

$$P_C : \operatorname{ri}(\operatorname{dom}\varphi) \to C\cap\operatorname{ri}(\operatorname{dom}\varphi).$$

It is evident that $P_C$ acts as the identity on $C\cap\operatorname{ri}(\operatorname{dom}\varphi)$, and it can be shown that $P_C$ is continuous.

There is also a variational characterization of the Bregman projection of a point $\boldsymbol{y}$ from ri(dom $\varphi$) onto the set $C$,

$$D_\varphi(\boldsymbol{x};\boldsymbol{y}) \geq D_\varphi(\boldsymbol{x};P_C(\boldsymbol{y})) + D_\varphi(P_C(\boldsymbol{y});\boldsymbol{y}) \qquad \text{for every } \boldsymbol{x}\in C\cap\operatorname{dom}\varphi. \tag{2.3}$$

Conversely, suppose we replace $P_C(\boldsymbol{y})$ with an arbitrary point $\boldsymbol{z}$ from $C\cap\operatorname{ri}(\operatorname{dom}\varphi)$ that verifies the inequality. Then $\boldsymbol{z}$ must indeed be the Bregman projection of $\boldsymbol{y}$ onto

$C$. When the constraint $C$ is an affine space (i.e., a translated subspace), then the Bregman projection of $\boldsymbol{y}$ onto $C$ has a formally stronger characterization,

$$D_\varphi(\boldsymbol{x};\boldsymbol{y}) = D_\varphi(\boldsymbol{x};P_C(\boldsymbol{y})) + D_\varphi(P_C(\boldsymbol{y});\boldsymbol{y}) \qquad \text{for every } \boldsymbol{x} \in C \cap \operatorname{dom}\varphi. \tag{2.4}$$

When the Bregman divergence is the Euclidean divergence, formula (2.3) reduces to the criterion for identifying the orthogonal projection onto a convex set [14, Chapter 4], while formula (2.4) is usually referred to as the Pythagorean theorem. These facts justify the assertion that Bregman projections generalize orthogonal projections.

When the constraint set $C$ and the Bregman divergence are simple enough, it may be possible to determine the Bregman projection onto $C$ analytically. For example, let us define the hyperplane $C = \{\boldsymbol{x} : \langle \boldsymbol{a}, \boldsymbol{x}\rangle = \alpha\}$. When $\|\boldsymbol{a}\|_2 = 1$, the projection of $\boldsymbol{y}$ onto $C$ with respect to the Euclidean divergence is

$$P_C(\boldsymbol{y}) = \boldsymbol{y} - (\langle \boldsymbol{a}, \boldsymbol{y}\rangle - \alpha)\, \boldsymbol{a}. \tag{2.5}$$

As a second example, suppose that $C$ contains a strictly positive vector and that $\boldsymbol{y}$ is strictly positive. Using Lagrange multipliers, we check that the projection of $\boldsymbol{y}$ onto $C$ with respect to the relative entropy has components

$$(P_C(\boldsymbol{y}))_n = y_n \exp\{\xi\, a_n\}, \quad \text{where } \xi \text{ is chosen so that } P_C(\boldsymbol{y}) \in C. \tag{2.6}$$

In the case when all the components of $\boldsymbol{a}$ are identical (to one, without loss of generality), then $\xi = \log\alpha - \log\sum_n y_n$.

It is uncommon that a Bregman projection can be explicitly determined. In section 3, we describe numerical methods for computing the Bregman projection onto a hyperplane, which is the foundation for producing Bregman projections onto more complicated sets. For another example of a projection that can be computed analytically, turn to the end of subsection 3.3.

**2.5. A cornucopia of divergences.** In this subsection, we will present some important Bregman divergences. The [illegible] divergences form the most fundamental class. A separable divergence arises from a seed function of the form

$$\varphi(\boldsymbol{x}) = \sum\nolimits_n w_n\, \varphi_n(x_n) \qquad \text{for positive weights } w_n.$$

If each $\varphi_n$ is Legendre, then the weighted sum is also Legendre. In the most common situation, the weights are constant and all the $\varphi_n$ are identical. In Table 2.1 we list some important Legendre functions on $\mathbb{R}$ that may be used to build separable divergences. These examples are adapted from [4] and [2]. Several of the divergences in Table 2.1 have names. We have already discussed the Euclidean divergence and the relative entropy. The bit entropy leads to a type of logistic loss, and the Burg entropy leads to the Itakura–Saito divergence.

Many of these univariate divergences are connected with well-known exponential families of probability distributions on $\mathbb{R}$. See Table 2.2 for some key examples drawn from [2].

One fundamental divergence is genuinely multidimensional. Suppose that $\boldsymbol{Q}$ is a positive-definite operator that acts on $\mathscr{X}$. We may construct a quadratic divergence on $\mathscr{X}$ from the seed function $\varphi(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{Q}\,\boldsymbol{x}, \boldsymbol{x}\rangle$, resulting in

$$D_\varphi(\boldsymbol{x};\boldsymbol{y}) = \tfrac{1}{2}\left\langle \boldsymbol{Q}\,(\boldsymbol{x}-\boldsymbol{y}), \boldsymbol{x}-\boldsymbol{y}\right\rangle.$$

Table 2.1
*Common seed functions and the corresponding divergences.*

| Function name | $\varphi(x)$ | $\operatorname{dom}\varphi$ | $D_\varphi(x;y)$ |
|---|---|---|---|
| Squared norm | $\frac{1}{2}x^2$ | $(-\infty,+\infty)$ | $\frac{1}{2}(x-y)^2$ |
| Shannon entropy | $x\log x - x$ | $[0,+\infty)$ | $x\log\frac{x}{y} - x + y$ |
| Bit entropy | $x\log x + (1-x)\log(1-x)$ | $[0,1]$ | $x\log\frac{x}{y} + (1-x)\log\frac{1-x}{1-y}$ |
| Burg entropy | $-\log x$ | $(0,+\infty)$ | $\frac{x}{y} - \log\frac{x}{y} - 1$ |
| Hellinger | $-\sqrt{1-x^2}$ | $[-1,1]$ | $(1-xy)(1-y^2)^{-1/2} - (1-x^2)^{1/2}$ |
| $\ell_p$ quasi-norm | $-x^p \qquad (0<p<1)$ | $[0,+\infty)$ | $-x^p + p\,xy^{p-1} - (p-1)\,y^p$ |
| $\ell_p$ norm | $\lvert x\rvert^p \qquad (1<p<\infty)$ | $(-\infty,+\infty)$ | $\lvert x\rvert^p - p\,x\operatorname{sgn}y\,\lvert y\rvert^{p-1} + (p-1)\,\lvert y\rvert^p$ |
| Exponential | $\exp x$ | $(-\infty,+\infty)$ | $\exp x - (x-y+1)\exp y$ |
| Inverse | $1/x$ | $(0,+\infty)$ | $1/x + x/y^2 - 2/y$ |

Table 2.2
*Common exponential families and the corresponding divergences.*

| Exponential family | $\psi(\theta)$ | $\operatorname{dom}\psi$ | $\mu(\theta)$ | $\varphi(x)$ | Divergence |
|---|---|---|---|---|---|
| Gaussian ($\sigma^2$ fixed) | $\frac{1}{2}\sigma^2\theta^2$ | $(-\infty,+\infty)$ | $\sigma^2\theta$ | $\frac{1}{2\sigma^2}x^2$ | Euclidean |
| Poisson | $\exp\theta$ | $(-\infty,+\infty)$ | $\exp\theta$ | $x\log x - x$ | Relative entropy |
| Bernoulli | $\log(1+\exp\theta)$ | $(-\infty,+\infty)$ | $\frac{\exp\theta}{1+\exp\theta}$ | $x\log x + (1-x)\log(1-x)$ | Logistic loss |
| Gamma ($\alpha$ fixed) | $-\alpha\,\log(-\theta)$ | $(-\infty,0)$ | $-\alpha/\theta$ | $-\alpha\log x + \alpha\log\alpha - \alpha$ | Itakura–Saito |

This divergence is connected to the exponential family of multivariate Gaussian distributions with covariance matrix $\boldsymbol{Q}^{-1}$. In the latter context, the square root of this divergence is often referred to as the [illegible] in statistics [29]. Other multidimensional examples arise when we compose the Euclidean norm with another function. For instance, one might consider the convex function $\varphi(\boldsymbol{x}) = -\sqrt{1 - \|\boldsymbol{x}\|_2^2}$ defined on the Euclidean unit ball. It yields the Hellinger-like divergence

$$D_\varphi(\boldsymbol{x};\boldsymbol{y}) = \frac{1 - \langle \boldsymbol{x}, \boldsymbol{y}\rangle}{\sqrt{1 - \|\boldsymbol{y}\|_2^2}} - \sqrt{1 - \|\boldsymbol{x}\|_2^2}.$$

**2.6. Matrix divergences.** Hermitian matrices admit a rich variety of divergences that were first studied in [4] using the methods of Lewis [27]. Let $\mathscr{H}$ be the space of $N \times N$ Hermitian matrices equipped with the real-linear inner product $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle = \operatorname{Re}\operatorname{Tr}\boldsymbol{X}\boldsymbol{Y}^*$. Define the function $\boldsymbol{\lambda} : \mathscr{H} \to \mathbb{R}^N$ that maps a Hermitian matrix to the vector listing its eigenvalues in algebraically decreasing order. Let $\varphi$ be a closed, proper, convex function on $\mathbb{R}^N$ that is invariant under coordinate permutation. That is, $\varphi(\boldsymbol{x}) = \varphi(\boldsymbol{P}\boldsymbol{x})$ for every permutation matrix $\boldsymbol{P}$.

By composing $\varphi$ with the eigenvalue map, we induce a real-valued function on Hermitian matrices. As the following theorem elaborates, the induced map has the same convexity properties as the function $\varphi$. Therefore, the induced map can be used as a seed function to define a Bregman divergence on the space of Hermitian matrices.

THEOREM 2 (Lewis [27, 26]). [illegible] $\varphi \circ \boldsymbol{\lambda}$ [illegible]

1. [illegible] $\varphi$ [illegible]
2. [illegible] $\varphi \circ \boldsymbol{\lambda}$ [illegible] $\boldsymbol{\lambda}$ [illegible] $\operatorname{dom}\varphi$
3. [illegible] $(\varphi \circ \boldsymbol{\lambda})^* = \varphi^* \circ \boldsymbol{\lambda}$
4. [illegible] $\boldsymbol{X}$ [illegible] $\varphi$ [illegible] $\boldsymbol{\lambda}(\boldsymbol{X})$ [illegible] $\boldsymbol{X}$ [illegible] $\boldsymbol{U}\{\operatorname{diag}\boldsymbol{\lambda}(\boldsymbol{X})\}\boldsymbol{U}^*$ [illegible]

$$\nabla(\varphi \circ \boldsymbol{\lambda})(\boldsymbol{X}) = \boldsymbol{U}\{\operatorname{diag}\nabla\varphi(\boldsymbol{\lambda}(\boldsymbol{X}))\}\boldsymbol{U}^*.$$

[illegible] $\varphi$ [illegible]

5. [illegible] $\varphi$ [illegible]

[illegible] $\varphi$ [illegible] $\varphi(\boldsymbol{x}) = \varphi(|\boldsymbol{x}|)$ [illegible] $\boldsymbol{x}$ [illegible] $\mathbb{R}^N$ [illegible] $|\cdot|$ [illegible]

Unitarily invariant matrix norms provide the most basic examples of induced maps. Indeed, item 3 of the last theorem generalizes von Neumann's famous result about dual norms of unitarily invariant prenorms [21, 438ff.].

An exquisite example of a matrix divergence arises from $\varphi(\boldsymbol{x}) = -\sum_n \log x_n$. The induced map is $(\varphi \circ \boldsymbol{\lambda})(\boldsymbol{X}) = -\log\det\boldsymbol{X}$, whose domain is the positive-definite cone. Since $\nabla(\varphi \circ \boldsymbol{\lambda})(\boldsymbol{X}) = -\boldsymbol{X}^{-1}$, the resulting divergence is

$$D_{\ell d}(\boldsymbol{X};\boldsymbol{Y}) = \left\langle \boldsymbol{X}, \boldsymbol{Y}^{-1}\right\rangle - \log\det\boldsymbol{X}\boldsymbol{Y}^{-1} - N. \tag{2.7}$$

Intriguingly, certain projections with respect to this divergence can be computed analytically. See subsection 3.3 for details.

Another important example arises from the negative Shannon entropy $\varphi(\boldsymbol{x}) = \sum_n x_n \log x_n - x_n$. The induced map is $(\varphi \circ \boldsymbol{\lambda})(\boldsymbol{X}) = \operatorname{Tr}(\boldsymbol{X} \log \boldsymbol{X} - \boldsymbol{X})$, whose domain is the positive-semidefinite cone. This matrix function arises in quantum mechanics, where it is referred to as the [illegible] [31]. It yields the divergence

$$D_{\mathrm{vN}}(\boldsymbol{X}; \boldsymbol{Y}) = \operatorname{Tr}[\boldsymbol{X}(\log \boldsymbol{X} - \log \boldsymbol{Y}) - \boldsymbol{X} + \boldsymbol{Y}], \tag{2.8}$$

which we will call the von Neumann divergence. In the quantum mechanics literature, this divergence is referred to as the [illegible] [31]. This formula does not literally hold if either matrix is singular, but a limit argument shows that the divergence is finite precisely when the null space of $\boldsymbol{X}$ contains the null space of $\boldsymbol{Y}$.

When the seed function $\varphi$ is separable, matrix divergences can be expressed in a way that emphasizes the distinct roles of the eigenvalues and eigenvectors. In particular, take $\varphi(\boldsymbol{x}) = \sum_n \varphi(x_n)$ and assume that $\boldsymbol{X}$ has eigenpairs $(\boldsymbol{u}_m, \mu_m)$ and that $\boldsymbol{Y}$ has eigenpairs $(\boldsymbol{v}_n, \nu_n)$. Then

$$\begin{aligned} D_{\varphi \circ \boldsymbol{\lambda}}(\boldsymbol{X}; \boldsymbol{Y}) &= \sum\nolimits_{m,n} |\langle \boldsymbol{u}_m, \boldsymbol{v}_n \rangle|^2 \left[\varphi(\mu_m) - \varphi(\nu_n) - \varphi'(\nu_n)(\mu_m - \nu_n)\right] \\ &= \sum\nolimits_{m,n} |\langle \boldsymbol{u}_m, \boldsymbol{v}_n \rangle|^2 D_\varphi(\mu_m; \nu_n). \end{aligned}$$

In words, the matrix divergence adds up the scalar divergences between pairs of eigenvalues, weighted by the squared cosine of the angle between the corresponding eigenvectors.

## 3. Computing Bregman projections.

It is not straightforward to compute the Bregman projection onto a general convex set. Unless additional structure is present, the best approach may be to apply standard convex optimization techniques. In this section, we discuss how to develop numerical methods for the basic problem of projecting onto a hyperplane or a halfspace. As we will see in sections 4 and 6, the projection onto an intersection of convex sets can be broken down into a sequence of projections onto the individual sets. Combining the two techniques, we can find the projection onto any affine space or polyhedral convex set.

### 3.1. Projection onto a hyperplane.

There is an efficient way to compute the Bregman projection onto a hyperplane. The key idea is to dualize the Bregman projection problem to obtain a nice one-dimensional problem. This approach can also be extended to produce the projection onto a halfspace because the convexity of the divergence implies that the projection lies on the boundary whenever the initial point is outside the halfspace.

We must solve the following convex program:

$$\min_{\boldsymbol{x}} D_\varphi(\boldsymbol{x}; \boldsymbol{y}) \qquad \text{subject to} \qquad \langle \boldsymbol{a}, \boldsymbol{x} \rangle = \alpha. \tag{3.1}$$

To ensure that this problem is well posed, we assume that $\operatorname{ri}(\operatorname{dom}\varphi)$ contains a feasible point. A necessary and sufficient condition on the solution $\boldsymbol{x}_\star$ of (3.1) is that the equation

$$\nabla_{\boldsymbol{x}} D_\varphi(\boldsymbol{x}; \boldsymbol{y}) = \xi \, \nabla_{\boldsymbol{x}} \left(\langle \boldsymbol{a}, \boldsymbol{x} \rangle - \alpha\right)$$

hold for a (unique) Lagrange multiplier $\xi \in \mathbb{R}$. The gradient of the divergence is $\nabla\varphi(\boldsymbol{x}) - \nabla\varphi(\boldsymbol{y})$, resulting in the equation

$$\nabla\varphi(\boldsymbol{x}_\star) = \xi \boldsymbol{a} + \nabla\varphi(\boldsymbol{y}).$$

The gradient of a Legendre function $\varphi$ is a bijection from $\operatorname{dom}\varphi$ to $\operatorname{dom}\varphi^*$, and its inverse is the gradient of the conjugate [35, Thm. 26.5]. Thus we obtain an explicit expression for the Bregman projection as a function of the unknown multiplier:

$$\boldsymbol{x}_\star = \nabla\varphi^*(\xi\boldsymbol{a} + \nabla\varphi(\boldsymbol{y})). \tag{3.2}$$

Form the inner product with $\boldsymbol{a}$ and enforce the constraint to reach

$$\langle \nabla\varphi^*(\xi\boldsymbol{a} + \nabla\varphi(\boldsymbol{y})), \boldsymbol{a}\rangle - \alpha = 0. \tag{3.3}$$

Now, the left-hand side of this equation is the derivative of the strictly convex, univariate function

$$J(\xi) = \varphi^*(\xi\boldsymbol{a} + \nabla\varphi(\boldsymbol{y})) - \alpha\xi. \tag{3.4}$$

There is an implicit constraint that the argument of $\varphi^*$ must lie within its domain. In view of (3.3), it becomes clear that the Lagrange multiplier is the unique minimizer of $J$. That is,

$$\xi_\star = \arg\min_\xi J(\xi).$$

Once we have determined the Lagrange multiplier, we introduce it into (3.2) to obtain the Bregman projection.

The best numerical method for minimizing $J$ depends strongly on the choice of the seed function $\varphi$. In some cases, the derivative(s) of $J$ may be difficult to evaluate. The second derivative may even fail to exist. To that end, we offer several observations that may be valuable.

1. The domain of $J$ contains a neighborhood of zero since $J(0) = \langle \boldsymbol{y}, \nabla\varphi(\boldsymbol{y})\rangle - \varphi(\boldsymbol{y})$.
2. Since $\varphi^*$ is a Legendre function, the first derivative of $J$ always exists. As shown in (3.3),

$$J'(\xi) = \langle \nabla\varphi^*(\xi\boldsymbol{a} + \nabla\varphi(\boldsymbol{y})), \boldsymbol{a}\rangle - \alpha.$$

3. When the Hessian of $\varphi^*$ exists, we have

$$J''(\xi) = \boldsymbol{a}^* \left\{\nabla^2\varphi^*(\xi\boldsymbol{a} + \nabla\varphi(\boldsymbol{y}))\right\} \boldsymbol{a}.$$

4. When the seed function $\varphi$ is separable, the Hessian $\nabla^2\varphi^*$ is diagonal.

The next two subsections provide examples that illustrate some of the issues involved in optimizing $J$.

**3.2. Example: Relative entropy.** Suppose that we wish to produce the Bregman projection of a nonnegative vector $\boldsymbol{y}$ onto the hyperplane $C = \{\boldsymbol{x} : \langle \boldsymbol{a}, \boldsymbol{x}\rangle = \alpha\}$ with respect to the relative entropy. This divergence arises from the seed function $\varphi(\boldsymbol{x}) = \sum_n x_n \log x_n - x_n$, whose conjugate is $\varphi^*(\boldsymbol{\theta}) = \sum_n \exp(\theta_n)$. To identify the Lagrange multiplier, we must minimize

$$J(\xi) = \sum\nolimits_n y_n \exp(\xi a_n) - \alpha\xi,$$

whose derivatives are

$$J'(\xi) = \sum\nolimits_n a_n y_n \exp(\xi a_n) - \alpha,$$

$$J''(\xi) = \sum\nolimits_n a_n^2 y_n \exp(\xi a_n).$$

These functions are all simple to evaluate, so it is best to use the Newton method preceded by a bracketing phase [32]. Once we have found the minimizer $\xi_\star$, the Bregman projection is

$$P_C(\boldsymbol{y}) = \boldsymbol{y} \cdot \exp(\xi_\star \boldsymbol{a}),$$

where $\cdot$ represents the Hadamard product and the exponential is performed componentwise.

**3.3. Example: Log-determinant divergence.** Here is a more sophisticated example that involves the log-determinant divergence. The divergence arises from the seed function $\varphi(\boldsymbol{X}) = -\log\det(\boldsymbol{X})$, whose domain is the positive-definite cone and whose gradient is $\nabla\varphi(\boldsymbol{X}) = -\boldsymbol{X}^{-1}$. The conjugate function $\varphi^*(\boldsymbol{\Theta}) = N - \log\det(-\boldsymbol{\Theta})$, whose domain is the negative-definite cone and whose gradient satisfies $\nabla\varphi^*(\boldsymbol{\Theta}) = -\boldsymbol{\Theta}^{-1}$.

Suppose we need to project the positive-definite matrix $\boldsymbol{Y}$ onto the hyperplane

$$C = \{\boldsymbol{X} : \langle \boldsymbol{A}, \boldsymbol{X}\rangle = \alpha\}, \qquad \text{where } \boldsymbol{A} = \boldsymbol{A}^*.$$

We must minimize

$$J(\xi) = N - \log\det(\boldsymbol{Y}^{-1} - \xi\boldsymbol{A}) - \alpha\xi,$$

while ensuring that $\boldsymbol{Y}^{-1} - \xi\boldsymbol{A}$ is positive definite.

Let $\boldsymbol{Y} = \boldsymbol{L}\boldsymbol{L}^*$, and abbreviate $\boldsymbol{W} = \boldsymbol{L}^*\boldsymbol{A}\boldsymbol{L}$, which is singular whenever $\boldsymbol{A}$ is rank deficient. Then the derivatives of $J$ can be expressed as

$$J'(\xi) = \operatorname{Tr}(\boldsymbol{W}(\mathbf{I} - \xi\boldsymbol{W})^{-1}) - \alpha,$$

$$J''(\xi) = \operatorname{Tr}\left( \left(\boldsymbol{W}(\mathbf{I} - \xi\boldsymbol{W})^{-1}\right)^2\right).$$

In general, $J$ and its derivatives are all costly. It appears that the most efficient way to calculate them for multiple values of the scalar $\xi$ is to preprocess $\boldsymbol{W}$ to extract its eigenvalues $\{\lambda_n\}$. It follows that

$$J'(\xi) = \left(\sum\nolimits_n \frac{\lambda_n}{1 - \lambda_n\xi}\right) - \alpha,$$

$$J''(\xi) = \sum\nolimits_n \left(\frac{\lambda_n}{1 - \lambda_n\xi}\right)^2.$$

It is worth cautioning that $\operatorname{dom} J = \{\xi : \xi < 1/\max_n \lambda_n\}$ since the matrix $\mathbf{I} - \xi\boldsymbol{W}$ must remain positive definite.

Once again, we see that a guarded or damped Newton method is the best way to optimize $J$. Given the solution $\xi_\star$, the Bregman projection is

$$P_C(\boldsymbol{Y}) = \boldsymbol{L}(\mathbf{I} - \xi_\star\boldsymbol{W})^{-1}\boldsymbol{L}^*.$$

We can reuse the eigenvalue decomposition to accelerate this final computation.

As shown in [25], these calculations simplify massively when the constraint matrix has rank one: $\boldsymbol{A} = \boldsymbol{a}\boldsymbol{a}^*$. In this case, we can find the zero of $J'$ analytically because $\boldsymbol{a}^*\boldsymbol{Y}\boldsymbol{a}$ is the only nonzero eigenvalue of $\boldsymbol{W}$. Then the Sherman–Morrison formula delivers an explicit expression for the projection:

$$P_C(\boldsymbol{Y}) = \boldsymbol{Y} + \frac{\boldsymbol{a}^*\boldsymbol{Y}\boldsymbol{a} - \alpha}{(\boldsymbol{a}^*\boldsymbol{Y}\boldsymbol{a})^2}(\boldsymbol{Y}\boldsymbol{a})(\boldsymbol{Y}\boldsymbol{a})^*.$$

The cost of performing the projection totals $O(N^2)$.

**4. The successive projection algorithm for affine constraints.** Now we describe an algorithm for solving (1.1) in the special case that the constraint sets are all [illegible]. In the next section, we will present some concrete problems to which this algorithm applies. The case of general convex constraint sets will be addressed afterward. We frame the following hypotheses.

| Assumption A.1 | |
|---|---|
| The divergence: | $\varphi$ is a convex function of Legendre type<br>$\operatorname{dom}\varphi^*$ is an open set |
| The constraints: | $C_1, C_2, \ldots, C_K$ are affine spaces with intersection $C$ |
| Constraint qualification: | $C \cap \operatorname{ri}(\operatorname{dom}\varphi)$ is nonempty |

Note that, by the results of subsection 2.3, all Bregman divergences that arise from regular exponential families satisfy Assumption A.1.

Given an input $\boldsymbol{y}_0$ from $\operatorname{ri}(\operatorname{dom}\varphi)$, we seek the Bregman projection of $\boldsymbol{y}_0$ onto the intersection $C$ of the affine constraints. In general, it may be difficult to produce $P_C(\boldsymbol{y}_0)$. Nevertheless, if the basic sets $C_1, \ldots, C_K$ are chosen well, it may be relatively straightforward to calculate the Bregman projection onto each basic set. This heuristic suggests an algorithm: Project successively onto each basic set in the hope that the sequence of iterates will converge to the Bregman projection onto the intersection. To make this approach work in general, it is clear that we must choose every set an infinite number of times, so we add one more requirement to Assumption A.1 as follows.

| Assumption A.2 | |
|---|---|
| The control mapping: | $r : \mathbb{N} \to \{1, \ldots, K\}$ is a sequence that takes each output value an infinite number of times |

Together, Assumptions A.1 and A.2 will be referred to as Assumption A. Here is a formal statement of the algorithm.

ALGORITHM A (successive projection). [illegible] A [illegible] $\boldsymbol{y}_0$ [illegible] $\operatorname{ri}(\operatorname{dom}\varphi)$ [illegible]

$$\boldsymbol{y}_t = P_{C_{r(t)}}(\boldsymbol{y}_{t-1}).$$

[illegible] $\{\boldsymbol{y}_t\}$ [illegible] $P_C(\boldsymbol{y}_0)$

We present a short proof that this algorithm is correct. We refer to the article [4] for the argument that the sequence converges, and we extend the elegant proof from [12] to show that the limit of the sequence yields the Bregman projection.

[illegible] Suppose that $\boldsymbol{a}$ is an arbitrary point in $C \cap \operatorname{dom}\varphi$. Since the seed function $\varphi$ is Legendre, Bregman projections with respect to the divergence fall in the relative interior of $\operatorname{dom}\varphi$. In particular, each iterate $\boldsymbol{y}_t$ belongs to $\operatorname{ri}(\operatorname{dom}\varphi)$. Therefore, we may apply the Pythagorean theorem (2.4) to see that

$$D_\varphi(\boldsymbol{a}; \boldsymbol{y}_{t-1}) = D_\varphi(\boldsymbol{a}; \boldsymbol{y}_t) + D_\varphi(\boldsymbol{y}_t; \boldsymbol{y}_{t-1}).$$

Observe that this equation defines a recurrence, which we may solve to obtain

$$D_\varphi(\boldsymbol{a}; \boldsymbol{y}_0) = D_\varphi(\boldsymbol{a}; \boldsymbol{y}_t) + \sum\nolimits_{i=1}^{t} D(\boldsymbol{y}_i; \boldsymbol{y}_{i-1}).$$

Under Assumption A, Theorem 8.1 of [4] shows that the sequence of iterates generated by Algorithm A converges to a point $\overline{\boldsymbol{y}}$ in $C \cap \operatorname{ri}(\operatorname{dom}\varphi)$. Since the divergence is continuous in its second argument, we may take limits to reach

$$D_\varphi(\boldsymbol{a}; \boldsymbol{y}_0) = D_\varphi(\boldsymbol{a}; \overline{\boldsymbol{y}}) + \sum\nolimits_{i=1}^{\infty} D_\varphi(\boldsymbol{y}_i; \boldsymbol{y}_{i-1}).$$

We chose $\boldsymbol{a}$ arbitrarily from $C \cap \operatorname{dom}\varphi$, so we may replace $\boldsymbol{a}$ by $\overline{\boldsymbol{y}}$ to see that the infinite sum equals $D_\varphi(\overline{\boldsymbol{y}}; \boldsymbol{y}_0)$. It follows that

$$D_\varphi(\boldsymbol{a}; \boldsymbol{y}_0) = D_\varphi(\boldsymbol{a}; \overline{\boldsymbol{y}}) + D_\varphi(\overline{\boldsymbol{y}}; \boldsymbol{y}_0).$$

This equation holds for each point $\boldsymbol{a}$ in $C \cap \operatorname{dom}\varphi$, so we see that $\overline{\boldsymbol{y}}$ meets the variational characterization (2.4) of $P_C(\boldsymbol{y}_0)$. Therefore, $\overline{\boldsymbol{y}}$ is the Bregman projection of $\boldsymbol{y}_0$ onto $C$. $\square$

If the sets $\{C_k\}$ are not affine, then Algorithm A will generally fail to produce the Bregman projection of $\boldsymbol{y}_0$ onto the intersection $C$. In section 6, we will discuss a more sophisticated iterative algorithm for solving this problem. Nevertheless, for general closed, convex constraint sets, the sequence of iterates generated by the successive projection algorithm still converges to a point in $C \cap \operatorname{ri}(\operatorname{dom}\varphi)$ [4, Theorem 8.1].

To obtain the convergence guarantee for Algorithm A, it may be necessary to work in an affine subspace of the ambient inner-product space. This point becomes important when computing the projections of nonnegative (as opposed to positive) vectors with respect to the relative entropy. It arises again when studying projections of rank-deficient matrices with respect to the von Neumann divergence. We will touch on this issue in subsections 5.1 and 5.3.

## 5. Examples with affine constraints.

This section presents three matrix nearness problems with affine constraints. The first requests the nearest contingency table with fixed marginals. A special case is to produce the nearest doubly stochastic matrix with respect to relative entropy. For this problem, the successive projection algorithm is identical to Kruithof's famous diagonal scaling algorithm [24, 13].

The second problem centers on a matrix nearness problem from data analysis, namely, that of finding matrix approximations based on the MBI principle, which is a generalization of Jaynes' maximum entropy principle [23].

The third problem shows how to construct the correlation matrix closest to a given positive-semidefinite matrix with respect to some matrix divergences. For reference, a correlation matrix is a positive-semidefinite matrix with a unit diagonal.

### 5.1. Contingency tables with fixed marginals.

A [illegible] is an array that exhibits the joint probability mass function of a collection of discrete random variables. A nonnegative rectangular matrix may be viewed as the contingency table for two discrete random variables. We will focus on this case since higher-dimensional contingency tables essentially are no more complicated.

Suppose that $p_{AB}$ is the joint probability mass function of two random variables $A$ and $B$ with sample spaces $\{1, 2, \ldots, M\}$ and $\{1, 2, \ldots, N\}$. We use $\boldsymbol{X}$ to denote the $M \times N$ contingency table whose entries are

$$x_{mn} = p_{AB}(A = m \text{ and } B = n).$$

A [illegible] of $p_{AB}$ is a linear function of $\boldsymbol{X}$. The most important marginals of $p_{AB}$ are the vector of row sums $\boldsymbol{X}\,\mathbf{e}$, which gives the distribution of $A$, and the vector of column sums $\mathbf{e}^T \boldsymbol{X}$, which gives the distribution of $B$. Here, $\mathbf{e}$ is a conformal vector of ones. The distribution of $A$ conditioned on $B = n$ is given by the $n$th column of $\boldsymbol{X}$, and the distribution of $B$ conditioned on $A = m$ is given by the $m$th row of $\boldsymbol{X}$.

However, we consider the more general case of arbitrary nonnegative matrices—we treat $\boldsymbol{X}$ as a member of the collection of $M \times N$ real matrices equipped with the inner product $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \operatorname{Tr} \boldsymbol{X}\boldsymbol{Y}^T$. Note that, for the above probabilistic interpretation, $\boldsymbol{X}$ must be scaled so that its entries sum to 1.

A common problem is to use an initial estimate to produce a contingency table that has fixed marginals. In this setting, nearness is typically measured with relative entropy

$$D(\boldsymbol{X}; \boldsymbol{Y}) = \sum\nolimits_{m,n} \left[ x_{mn} \log \frac{x_{mn}}{y_{mn}} - x_{mn} + y_{mn} \right].$$

An important special case is to find the doubly stochastic matrix nearest to a nonnegative square matrix $\boldsymbol{Y}_0$. In this case, we have two constraint sets

$$C_1 = \{\boldsymbol{X} : \boldsymbol{X}\,\mathbf{e} = \mathbf{e}\} \qquad \text{and} \qquad C_2 = \{\boldsymbol{X} : \mathbf{e}^T \boldsymbol{X} = \mathbf{e}^T\}.$$

It is clear that the intersection $C = C_1 \cap C_2$ contains the set of doubly stochastic matrices. In fact, every nonnegative matrix in $C$ is doubly stochastic. Using (2.6), it is easy to see that Bregman projection of a matrix onto $C_1$ with respect to the relative entropy is accomplished by rescaling the rows so that each row sums to one. Likewise, Bregman projection of a matrix onto $C_2$ is accomplished by rescaling the columns. Beginning with $\boldsymbol{Y}_0$, the successive projection algorithm alternately rescales the rows and columns. This procedure, of course, is the diagonal scaling algorithm of Kruithof [24, 13], sometimes called Sinkhorn's algorithm [36]. Our approach yields a geometric interpretation of the algorithm as a method for solving a matrix nearness problem by alternating Bregman projections. It is interesting that the nonnegativity constraint is implicitly enforced by the domain of the relative entropy. This viewpoint can be traced to the work of Ireland and Kullback [22].

There is still a subtlety that requires attention. Assumption A apparently requires that $C$ contain a matrix with strictly positive entries and that the input matrix $\boldsymbol{Y}_0$ be strictly positive. In fact, we may relax these premises. A nonnegative matrix whose zero pattern does not cover the zero pattern of $\boldsymbol{Y}_0$ has an infinite divergence from $\boldsymbol{Y}_0$. Therefore, we may as well restrict our attention to the linear space of matrices whose zero pattern covers that of $\boldsymbol{Y}_0$. Now we see that the constraint qualification in Assumption A requires that $C$ contain a matrix with exactly the same zero pattern as $\boldsymbol{Y}_0$. If it does, the algorithm will still converge to the Bregman projection of $\boldsymbol{Y}_0$ onto the doubly stochastic matrices. Determining whether the constraint qualification holds will generally involve a separate investigation [30].

It is also worth noting that Algorithm A encompasses other iterative methods for scaling to doubly stochastic form. At each step, for example, one might rescale only the row or column whose sum is most inaccurate. Parlett and Landis have considered algorithms of this sort [33]. The problem of scaling to have other row and column sums also fits neatly into our framework, and it has the same geometric interpretation.

**5.2. MBI and matrix approximation.** This section discusses a novel matrix nearness problem that arises in data analysis. Given a collection of vectors $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\} \subset \operatorname{dom} \varphi$, the [illegible] [2] of the collection is defined to be

$$I_\varphi(X) = \sum\nolimits_{j=1}^{N} w_j \, D_\varphi(\boldsymbol{x}_j; \boldsymbol{\mu}), \tag{5.1}$$

where $w_1, w_2, \ldots, w_N$ are nonnegative weights that sum to one, and $\boldsymbol{\mu}$ is the (weighted) arithmetic mean of the collection, i.e., $\boldsymbol{\mu} = \sum_j w_j \, \boldsymbol{x}_j$. Bregman information generalizes the notion of the [illegible], $\sigma^2 = N^{-1} \sum_j \|\boldsymbol{x}_j - \boldsymbol{\mu}\|_2^2$, of a Gaussian random variable (where each $w_j = N^{-1}$). When $D_\varphi$ is the relative entropy, the Bregman information that arises with an appropriate choice of weights is called [illegible], a fundamental quantity in information theory [11].

Bregman information exhibits an interesting connection with Jensen's inequality for a convex function $\varphi$:

$$\sum\nolimits_j w_j \, \varphi(\boldsymbol{x}_j) \geq \varphi\left(\sum\nolimits_j w_j \, \boldsymbol{x}_j\right).$$

Substituting $\boldsymbol{\mu} = \sum_j w_j \, \boldsymbol{x}_j$, we see that the difference between the two sides of the foregoing relation satisfies

$$\begin{aligned} \sum\nolimits_j w_j \, \varphi(\boldsymbol{x}_j) - \varphi(\boldsymbol{\mu}) &= \sum\nolimits_j w_j \, \varphi(\boldsymbol{x}_j) - \varphi(\boldsymbol{\mu}) - \left\langle \nabla\varphi(\boldsymbol{\mu}), \sum\nolimits_j w_j \boldsymbol{x}_j - \boldsymbol{\mu} \right\rangle \\ &= \sum\nolimits_j w_j \left[\varphi(\boldsymbol{x}_j) - \varphi(\boldsymbol{\mu}) - \langle \nabla\varphi(\boldsymbol{\mu}), \boldsymbol{x}_j - \boldsymbol{\mu}\rangle\right] \\ &= I_\varphi(X). \qquad (5.2) \end{aligned}$$

In words, the Bregman information is the disparity between the two sides of Jensen's inequality. Equation (5.2) can also be viewed as a generalization of the relationship between the variance and the arithmetic mean,

$$\sigma^2 = N^{-1} \sum\nolimits_j \|\boldsymbol{x}_j\|_2^2 - \|\boldsymbol{\mu}\|_2^2.$$

Let us describe an application of Bregman information in data analysis. In this field, matrix approximations play a central role. Unfortunately, many common approximations destroy essential structure in the data matrix. For example, consider the $k$-truncated singular value decomposition (TSVD), which provides the best rank-$k$ Frobenius-norm approximation of a matrix. In information retrieval applications, however, the matrix that describes the co-occurrence of words and documents is both sparse and nonnegative. The TSVD ruins both of these properties. In this setting, the Frobenius norm is meaningless; relative entropy is the correct divergence measure according to the unigram or multinomial language model.

We may also desire that the matrix approximation satisfy some additional constraints. For instance, it may be valuable for the approximation to preserve marginals (i.e., linear functions) of the matrix entries. Let us formalize this idea. Suppose that $\boldsymbol{Y}$ is an $M \times N$ data matrix. We seek an approximation $\widetilde{\boldsymbol{X}}$ that satisfies the constraints

$$C_k = \{\boldsymbol{X} : \langle \boldsymbol{X}, \boldsymbol{A}_k\rangle = \langle \boldsymbol{Y}, \boldsymbol{A}_k\rangle\} \qquad k = 1, \ldots, K,$$

where each $\boldsymbol{A}_k$ is a fixed constraint matrix. We will write $C = \bigcap_k C_k$. As an example, $\boldsymbol{X}$ can be required to preserve the row and/or column sums of $\boldsymbol{Y}$.

Many different matrices, including the original matrix $\boldsymbol{Y}$, may satisfy these constraints. Clearly, a good matrix approximation should involve some reduction in the number of parameters used to represent the matrix. The key question is to decide how to produce the right approximation from $C$. One rational approach invokes the principle of [illegible] (MBI) [1], which states that the approximation should be the (unique) solution of the problem

$$\min_{\boldsymbol{X} \in C} I_\varphi(\boldsymbol{X}) = \min_{\boldsymbol{X} \in C} \sum\nolimits_{m,n} w_{mn} \, D_\varphi(x_{mn}, \mu), \qquad (5.3)$$

where $w_{mn}$ are prespecified weights and $\mu = \sum_{m,n} w_{mn} x_{mn}$. If the weights $w_{mn}$ and the matrix entries $x_{mn}$ are both sets of nonnegative numbers that sum to one, and if the Bregman divergence is the relative entropy, then the MBI principle reduces to Jaynes' maximum entropy principle [23]. Thus, the MBI principle tries to obtain as uniform an approximation as possible subject to the specified constraints. Note that problem (5.3) can be readily solved by the successive projection algorithm.

Next, we consider an important and natural source of constraints. [illegible] is the problem of partitioning a set of objects into clusters, where each cluster contains "similar" objects. Data matrices often capture the relationships between two sets of objects, such as word–document matrices in information retrieval and gene-expression matrices in bioinformatics. In such applications, it is often desirable to solve the coclustering problem, i.e., to simultaneously cluster the rows and columns of a data matrix. Formally, a co-clustering $(\rho, \gamma)$ is a partition of the rows into $I$ row clusters $\rho_1, \ldots, \rho_I$ and the columns into $J$ column clusters $\gamma_1, \ldots, \gamma_J$, i.e.,

$$\bigcup_{i=1}^{I} \rho_i = \{1, 2, \ldots, M\}, \qquad \text{where} \qquad \rho_i \cap \rho_\ell = \emptyset \quad \text{for } i \neq \ell,$$

$$\bigcup_{j=1}^{J} \gamma_j = \{1, 2, \ldots, N\}, \qquad \text{where} \qquad \gamma_j \cap \gamma_\ell = \emptyset \quad \text{for } j \neq \ell.$$

Given a coclustering, the rows belonging to row cluster $\rho_1$ can be arranged first, followed by rows belonging to row cluster $\rho_2$, etc. Similarly the columns can be re-ordered. This re-ordering has the effect of dividing the matrix into $I \cdot J$ subblocks, each of which is called a [illegible].

The coclustering problem is to search for the "best" possible row and column clusters. One way to measure the quality of a coclustering is to associate it with its MBI matrix approximation. A natural constraint set $C_{(\rho,\gamma)}$ for the coclustering problem contains matrices that preserve marginals of all the $I \cdot J$ coclusters (local information) in addition to row and column marginals (global information). With this constraint set, a formal objective for the coclustering problem is to find $(\rho, \gamma)$, which corresponds to the best possible MBI approximation:

$$\min_{\rho,\gamma} D_\varphi(\boldsymbol{Y}; \boldsymbol{X}_{(\rho,\gamma)}), \qquad \text{where} \qquad \boldsymbol{X}_{(\rho,\gamma)} = \arg \min_{\boldsymbol{X} \in C_{(\rho,\gamma)}} I_\varphi(\boldsymbol{X}). \tag{5.4}$$

This formulation yields an optimal coclustering as well as its associated MBI matrix approximation. The quality of such matrix approximations is a topic for further study. Note that problem (5.4) requires a combinatorial search, and it is known to be NP-complete. The most familiar clustering formulation, namely, the $k$-means problem, is the special case of (5.4) obtained from the Euclidean divergence, the choice $J = N$, and the condition of preserving cocluster sums.

As an example, consider the nonnegative matrix

$$\boldsymbol{Y} = \begin{bmatrix} 5 & 5 & 5 & 0 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 5 & 5 \\ 0 & 0 & 0 & 5 & 5 & 5 \\ 4 & 4 & 0 & 4 & 4 & 4 \\ 4 & 4 & 4 & 0 & 4 & 4 \end{bmatrix}.$$

On using coclustering (three row clusters and two column clusters), preserving row sums, column sums, and cocluster sums, the MBI principle (with relative entropy as the Bregman divergence) yields the matrix approximation

$$\boldsymbol{X}_1 = \begin{bmatrix} 5.4 & 5.4 & 4.2 & 0 & 0 & 0 \\ 5.4 & 5.4 & 4.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.2 & 5.4 & 5.4 \\ 0 & 0 & 0 & 4.2 & 5.4 & 5.4 \\ 3.6 & 3.6 & 2.8 & 2.8 & 3.6 & 3.6 \\ 3.6 & 3.6 & 2.8 & 2.8 & 3.6 & 3.6 \end{bmatrix}.$$

Note that this approximation has rank two and preserves nonnegativity as well as most of the nonzero structure of $\boldsymbol{Y}$. It can be verified that all the cocluster sums, row sums, and column sums of $\boldsymbol{X}_1$ match those of $\boldsymbol{Y}$. In contrast, the rank-two SVD approximation

$$\boldsymbol{X}_2 = \begin{bmatrix} 5.09 & 5.09 & 4.66 & -0.69 & 0.29 & 0.29 \\ 5.09 & 5.09 & 4.66 & -0.69 & 0.29 & 0.29 \\ 0.29 & 0.29 & -0.69 & 4.66 & 5.09 & 5.09 \\ 0.29 & 0.29 & -0.69 & 4.66 & 5.09 & 5.09 \\ 3.04 & 3.04 & 1.98 & 3.51 & 4.41 & 4.41 \\ 4.41 & 4.41 & 3.51 & 1.98 & 3.04 & 3.04 \end{bmatrix}$$

preserves neither the nonnegativity, the nonzero structure, nor the marginals of $\boldsymbol{Y}$.

**5.3. The nearest correlation matrix.** A [illegible] is a (real) positive-semidefinite matrix with a unit diagonal. Correlation matrices arise in statistics and applications such as finance, where they display the normalized second-order statistics (i.e., pairwise correlation coefficients) of a collection of random variables. In the deterministic setting, a correlation matrix may be viewed as the Gram matrix of a collection of unit vectors.

Higham has recently studied the nearest correlation matrix problem measuring distances using a type of weighted Frobenius norm [19]. Higham solves the problem by means of the Dykstra–Han algorithm given in section 6, alternating between the positive-semidefinite cone and the set of matrices with unit diagonal. We have observed that the nearest correlation matrix problem can be posed with Bregman divergences and, in particular, with matrix divergences.

Let us consider the problem of producing the correlation matrix closest to a given positive-semidefinite matrix with respect to the von Neumann divergence

$$D_{\mathrm{vN}}(\boldsymbol{X};\boldsymbol{Y}) = \mathrm{Tr}\,[\boldsymbol{X}(\log\boldsymbol{X} - \log\boldsymbol{Y}) - \boldsymbol{X} + \boldsymbol{Y}].$$

In case $\boldsymbol{Y}$ is singular, we must restrict our attention to the linear space of matrices whose null space contains the null space of $\boldsymbol{Y}$. After taking this step, one must interpret the formulae with care. These remarks signal our reason for employing the von Neumann divergence to measure the disparity between correlation matrices. A matrix $\boldsymbol{X}$ has an infinite divergence from $\boldsymbol{Y}$ unless the null space of $\boldsymbol{X}$ contains the null space of $\boldsymbol{Y}$. In particular, the rank of the Bregman projection of $\boldsymbol{Y}$ onto the correlation matrices cannot exceed the rank of $\boldsymbol{Y}$. See also the examples at the end of this subsection.

The correlation matrices can be viewed as the intersection of the set of unit-diagonal matrices with the positive-semidefinite cone. This cone is also the domain of the von Neumann divergence, so we do not need to explicitly enforce the positive-semidefinite constraint. In fact, we need only project onto the set $C$ of matrices whose diagonal entries all equal one. It is natural to view $C$ as the intersection of the affine constraint sets

$$C_k = \{\boldsymbol{X} : x_{kk} = 1\}.$$

There is no explicit formula for the projection of a matrix $\boldsymbol{Y}$ onto the set $C_k$, but the discussion in section 3 shows that we can solve the problem by minimizing the function (3.4), which, in this example, reads

$$J(\xi) = \operatorname{Tr} \exp\{\log \boldsymbol{Y} + \xi\, \mathbf{e}_k \mathbf{e}_k^T\} - \xi, \tag{5.5}$$

where $\mathbf{e}_k$ is the $k$th canonical basis vector. Given the minimizer $\xi_\star$, the projection of $\boldsymbol{Y}$ onto $C_k$ is

$$P_{C_k}(\boldsymbol{Y}) = \exp\{\log \boldsymbol{Y} + \xi_\star\, \mathbf{e}_k \mathbf{e}_k^T\}. \tag{5.6}$$

Beware that one cannot read these formulae literally when $\boldsymbol{Y}$ is rank deficient! In any case, the numerical calculations are not trivial to perform. In order to apply the Newton method, the second derivative of $J$ is needed, which is more involved due to the noncommutativity of matrix multiplication.

Unfortunately, treating these issues in detail is beyond the scope of this paper.

There is an interesting special case that can be treated without optimization: the von Neumann projection of a matrix with constant diagonal onto the correlation matrices can always be obtained by rescaling. In particular, the projection preserves the zero pattern of the matrix and the eigenvalue distribution. To verify this point, suppose the diagonal entries of $\boldsymbol{Y}$ equal $\alpha$, and set $\boldsymbol{X} = \alpha^{-1}\boldsymbol{Y}$. According to the Karush–Kuhn–Tucker conditions, $\boldsymbol{X}$ is the Bregman projection of $\boldsymbol{Y}$ onto the set $C$ provided that $\nabla_{\boldsymbol{X}} D_{\text{vN}}(\boldsymbol{X}; \boldsymbol{Y})$ is diagonal. The latter gradient equals $\log \boldsymbol{X} - \log \boldsymbol{Y} + \mathbf{I}$, and a short calculation completes the argument. In contrast, the Frobenius norm projection of a matrix with constant diagonal does not preserve its nonzero structure or eigenvalue distribution. As an example, let $\boldsymbol{Y}$ be the $4 \times 4$ symmetric tridiagonal Toeplitz matrix with 2's on the diagonal and $-1$'s on the off-diagonal. The nearest correlation matrix to it, in the Frobenius norm, equals (to the figures shown)

$$\begin{bmatrix} 1.0000 & -0.8084 & 0.1916 & 0.1068 \\ -0.8084 & 1.0000 & -0.6562 & 0.1916 \\ 0.1916 & -0.6562 & 1.0000 & -0.8084 \\ 0.1068 & 0.1916 & -0.8084 & 1.0000 \end{bmatrix}.$$

As a second example, draw a random orthogonal matrix $\boldsymbol{Q}$ and form the rank-deficient matrix $\boldsymbol{Y} = \boldsymbol{Q}\, \operatorname{diag}(1, 10^{-3}, 10^{-6}, 0)\, \boldsymbol{Q}^T$. For instance,

$$\boldsymbol{Y} = \begin{bmatrix} .18335 & -.15180 & .08258 & -.34620 \\ -.15180 & .12606 & -.06887 & .28655 \\ .08258 & -.06887 & .03786 & -.15582 \\ -.34620 & .28655 & -.15582 & .65373 \end{bmatrix}.$$

The correlation matrix nearest to $\boldsymbol{Y}$ in the Frobenius norm is obtained by simply shifting the diagonal:

$$\boldsymbol{X}_1 = \begin{bmatrix} 1.0000 & -.15180 & .08258 & -.34620 \\ -.15180 & 1.0000 & -.06887 & .28655 \\ .08258 & -.06887 & 1.0000 & -.15582 \\ -.34620 & .28655 & -.15582 & 1.0000 \end{bmatrix}.$$

Meanwhile, the nearest correlation matrix with respect to the von Neumann divergence has the same range space as $\boldsymbol{Y}$ and thus is also of rank 3:

$$\boldsymbol{X}_2 = \begin{bmatrix} 1.0000 & -.77271 & .59020 & -.99995 \\ -.77271 & 1.0000 & -.96847 & .77080 \\ .59020 & -.96847 & 1.0000 & -.58778 \\ -.99995 & .77080 & -.58778 & 1.0000 \end{bmatrix}.$$

Note that due to space limitations, all of the above matrices are shown only to five digits of accuracy. The eigenvalues of $\boldsymbol{X}_1$ are 0.611, 0.851, 0.951, and 1.588, while the nonzero eigenvalues of $\boldsymbol{X}_2$ are $0.457 \times 10^{-6}, 0.650 \times 10^{-2}$, and 3.350. Thus we see that the Frobenius norm solution does not preserve the small eigenvalues, while the von Neumann divergence solution preserves the rank and also tries to preserve the eigenvalue distribution.

The recent literature contains a substantial amount of work on numerical methods for calculating nearest correlation matrices with respect to the Frobenius norm. Higham describes an alternating projection method, as well as an approach via semidefinite programming [19]. Malick [28] and Boyd and Xiao [7] study efficient algorithms for solving the dual of a more general projection problem, while Qi and Sun [34] develop a generalized Newton method for the nearest correlation matrix problem.

In contrast, the problem of finding nearest correlation matrices with respect to a Bregman divergence is virtually unstudied. The main motivation for studying this problem is that it leads to correlation matrices that have a very different character, which may be more appropriate in applications. For example, as shown above, the method for solving the von Neumann nearness problem may yield low-rank correlation matrices. This type of solution has immense practical value because it explains the data using a small number of [illegible] [17]. In contrast, the Frobenius norm solution may increase the rank of the matrix. Unfortunately, to apply our technique, the initial matrix must lie in the domain of the von Neumann divergence, i.e., the positive-semidefinite cone. One remedy is to preprocess the matrix by performing a Frobenius projection onto the positive-semidefinite cone.

The broad scope of the present article limits the amount of detail we can provide, so we have been only able to sketch one algorithm for solving the nearest correlation matrix problem. It would be valuable to devise more powerful algorithms by invoking ideas from the papers cited above.

## 6. The successive projection–correction algorithm for convex constraints.

This section describes an algorithm for solving the Bregman nearness problem (1.1) in the case where the constraints are closed, convex sets. In the succeeding section, we will present some nearness problems to which this algorithm applies. We frame the following hypotheses:

| Assumption B | |
|---|---|
| The divergence: | $\varphi$ is a convex function of Legendre type<br>$\varphi$ is cofinite, i.e., $\operatorname{dom}\varphi^* = \mathscr{X}$ |
| The constraints: | $C_1, \ldots, C_K$ are closed, convex sets with intersection $C$ |
| Constraint qualification: | $\operatorname{ri}(C_1) \cap \cdots \cap \operatorname{ri}(C_K) \cap \operatorname{ri}(\operatorname{dom}\varphi)$ is nonempty |
| The control mapping: | $r : \mathbb{N} \to \{1, 2, \ldots, K\}$ is a sequence that takes each output value at least once during each $T$ consecutive input values |

Given an input $\boldsymbol{y}_0$ from $\operatorname{ri}(\operatorname{dom}\varphi)$, we seek the Bregman projection of $\boldsymbol{y}_0$ onto $C$ with respect to the divergence $D_\varphi$. As before, the algorithm projects successively onto each constraint set. Since the sets are no longer affine, it is also necessary to introduce a correction term to guide the algorithm toward the Bregman projection. This algorithm generalizes a method for the Euclidean divergence that was developed independently by Dykstra [16] and Han [18].

Algorithm B (successive projection–correction). [illegible] B [illegible] $\boldsymbol{y}_0 \in \operatorname{ri}(\operatorname{dom}\varphi)$ [illegible]

1. [illegible] $\boldsymbol{q}^k = \boldsymbol{0}$ [illegible] $k = 1, \ldots, K$
2. [illegible]

$$\boldsymbol{y}_{t+1} \leftarrow P_{C_{r(t)}}\left(\nabla\varphi^*\left(\nabla\varphi(\boldsymbol{y}_t) + \boldsymbol{q}^{r(t)}\right)\right).$$

3. [illegible]

$$\boldsymbol{q}^{r(t)} \leftarrow \boldsymbol{q}^{r(t)} + \nabla\varphi(\boldsymbol{y}_t) - \nabla\varphi(\boldsymbol{y}_{t+1}).$$

4. [illegible] 2.

[illegible] $\boldsymbol{y}_0$ [illegible] $C$ [illegible] $D_\varphi$

$$P_C(\boldsymbol{y}_0) = \lim_{t\to\infty} \boldsymbol{y}_t.$$

The proof that this algorithm succeeds is quite burdensome, and none of the arguments in the literature are especially intuitive. The correctness of the algorithm that we have presented here follows from Tseng's general framework [37]. His paper contains only the development for the Euclidean divergence; see [6, 9] for comments on the extension. The literature contains several other proofs with somewhat different hypotheses [6, 9]. We feel that the above version offers the best tradeoff between applicability and accessibility.

The following connections may help the reader understand the algorithm somewhat better. It is possible to identify this procedure as a generalization of Bregman's algorithm for minimizing strictly convex functions [10, 9]. Bregman's algorithm is a primal-dual method that maximizes with respect to one dual variable (the $\boldsymbol{q}^k$) at a time, while maintaining the Karush–Kuhn–Tucker conditions on the primal problem. It is also possible to view the algorithm as a coordinate ascent algorithm for an optimization problem that is dual to the projection problem [37]. It is for this reason that the update in step 2 closely resembles the dual function $J$ obtained in section 3.

### 6.1. Comparing the algorithms.

Let us take a moment to weigh the successive projection–correction algorithm (Algorithm B) against the successive projection algorithm (Algorithm A). It is most important to note that Algorithm A applies only to the case where the constraints are affine, while Algorithm B succeeds for general closed, convex constraints. It can be shown that the corrections in Algorithm B are unnecessary when the constraints are affine, so it reduces to Algorithm A [9].

Although it may appear that Algorithm A has a weaker constraint qualification, the difference here is purely formal. We remark that the constraint qualification in Algorithm B can be weakened when some of the constraint sets are polyhedral, i.e., can be written as a finite intersection of halfspaces. In that case, we may remove the relative interior from the polyhedral constraint sets in the constraint qualification.

The methods also place different hypotheses on the divergence; Algorithm B asks more from the seed function $\varphi$ than Algorithm A. The former requires that $\operatorname{dom} \varphi^* = \mathscr{X}$ while the latter needs only $\operatorname{dom} \varphi^*$ to be open. For example, the Burg entropy $\varphi(x) = -\log(x)$ is admissible for Algorithm A but not for Algorithm B.

Finally, the control mapping for Algorithm B is more restrictive than the control mapping for Algorithm A. The former requires that the projections be performed in almost cyclic order, while the latter requires only that each constraint set should appear an infinite number of times.

## 7. Examples with convex constraints.

This section discusses two matrix nearness problems that involve nonaffine constraints. First, we discuss the *metric nearness problem*, which elicits the closest metric graph to a given weighted graph. We have already studied this problem with respect to norms in [15]. Here, we expand our treatment to Bregman divergences.

Second, we study an important problem in data analysis, namely learning a so-called "kernel" or similarity matrix that satisfies constraints that arise from knowledge of the underlying application domain.

### 7.1. The metric nearness problem.

We recently encountered a striking new matrix nearness problem [15] while studying an application in computational biology. In this article, we extend the problem to Bregman divergences and show that it can be solved using the successive projection–correction algorithm (Algorithm B).

Suppose that $\boldsymbol{X}$ is the adjacency matrix of an undirected, weighted graph on $N$ vertices. That is, $x_{mn}$ registers the weight of the edge between vertices $m$ and $n$. Since the graph is undirected, $\boldsymbol{X}$ is a symmetric matrix. We will also assume that $\boldsymbol{X}$ is *hollow* (i.e., has a zero diagonal). If one interprets the weights as distances, it is natural to ask whether the graph can be embedded in a metric space. Indeed, the embedding is possible if and only if the triangle inequalities hold, i.e.,

$$x_{mn} \leq x_{m\ell} + x_{\ell n} \qquad \text{for each triple of distinct vertices } (\ell, m, n). \tag{7.1}$$

Note that the condition (7.1) implies that the weights are nonnegative, provided that $\boldsymbol{X}$ is symmetric. We will refer to any hollow, symmetric matrix that satisfies (7.1) as a *metric adjacency matrix*.

The *metric nearness problem* is to find the metric adjacency matrix closest to a given adjacency matrix. We view this nearness problem as an agnostic method for learning a metric from noisy distance measurements. It is entirely distinct from multidimensional scaling, which requests an ensemble of points in a [illegible] metric space (usually Euclidean) that realizes a given set of distances. In our first report on this problem [15], we used weighted matrix norms to measure the distance between adjacency matrices. In this article, we will use Bregman divergences. Note that the divergence is unrelated to the metric encoded in the entries of the adjacency matrix; the divergence is used to determine how much one adjacency matrix (i.e., graph) differs from another.

By this point, it should be clear how we propose to solve the metric nearness problem. We will work in the space of hollow, symmetric matrices. It is evident that the metric adjacency matrices from a closed, convex cone $C$. Clearly, $C$ is the intersection of $\binom{N}{3}$ halfspaces:

$$C_{\ell mn} = \{\boldsymbol{X} : x_{mn} - x_{m\ell} - x_{\ell n} \leq 0\},$$

where $\ell$, $m$, and $n$ index distinct vertices. Therefore, we may apply Algorithm B.

To be concrete, we will consider Bregman projections with respect to the relative entropy. For reference, the seed function is

$$\varphi(\boldsymbol{X}) = \sum\nolimits_{mn} \left[x_{mn} \log x_{mn} - x_{mn}\right],$$

which has Fenchel conjugate

$$\varphi^*(\boldsymbol{Y}) = \sum\nolimits_{mn} \exp y_{mn}.$$

The divergence is

$$D_\varphi(\boldsymbol{X}; \boldsymbol{Y}) = \sum\nolimits_{mn} \left[x_{mn} \log \frac{x_{mn}}{y_{mn}} - x_{mn} + y_{mn}\right].$$

This divergence has an interesting advantage over the Frobenius norm. If the original adjacency matrix does not contain zero distances, then the projection on the metric adjacency matrices will not contain any zero distances. This fact ensures that the final matrix defines a genuine metric, rather than a pseudometric.

Algorithm B requires that we compute the Bregman projection of a matrix that has the form $\boldsymbol{X} = \nabla\varphi^*(\nabla\varphi(\boldsymbol{Y}_t) + \boldsymbol{Q}^{\ell mn})$, where $\boldsymbol{Q}^{\ell mn}$ is a dual variable. It is easy to check that this expression reduces to

$$\boldsymbol{X} = \boldsymbol{Y}_t \cdot \exp\cdot(\boldsymbol{Q}^{\ell mn}),$$

where $\cdot$ is the Hadamard (i.e., componentwise) product and $\exp\cdot$ is the Hadamard exponential. We will see that the dual variable $\boldsymbol{Q}^{\ell mn}$ has at most six nonzero entries. Therefore, the matrix $\boldsymbol{X}$ differs from $\boldsymbol{Y}_t$ in at most six places.

It is straightforward to calculate the Bregman projection $\boldsymbol{Y}_{t+1}$ of the matrix $\boldsymbol{X}$ onto the constraint $C_{\ell mn}$. If $\boldsymbol{X}$ already falls in the constraint set, then the projection $\boldsymbol{Y}_{t+1} = \boldsymbol{X}$. Otherwise, set $\delta = \sqrt{(x_{m\ell} + x_{\ell n})/x_{mn}}$. The entries of the projection $\boldsymbol{Y}_{t+1}$ are identical to those of $\boldsymbol{X}$ except for the following six:

$$\begin{aligned} y_{mn} &= \delta\, x_{mn} & \qquad y_{nm} &= y_{mn} \\ y_{m\ell} &= x_{m\ell}/\delta & \qquad y_{\ell m} &= y_{m\ell} \\ y_{\ell n} &= x_{\ell n}/\delta & \qquad y_{n\ell} &= y_{\ell n}. \end{aligned}$$

In words, the projection determines how much the triangle inequality is violated, and it distributes the deficit multiplicatively among the three edges.

Finally, the algorithm updates the dual variable $\boldsymbol{Q}^{\ell mn}$ associated with the constraint using the formula

$$\boldsymbol{Q}^{\ell mn} \leftarrow \boldsymbol{Q}^{\ell mn} + \log\cdot(\boldsymbol{Y}_t) - \log\cdot(\boldsymbol{Y}_{t+1})$$

where $\log\cdot$ is the Hadamard logarithm. This update affects only six entries of $\boldsymbol{Q}^{\ell mn}$. In practice, we would store only the upper triangle of the adjacency matrices, so the update touches only three entries.

Consider the following adjacency matrix, which fails to be a metric graph,

$$\boldsymbol{Y} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 10000 & 1 & 1 & 1 \\ 1 & 10000 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

The nearest metric adjacency matrix in relative entropy is found to be

$$\boldsymbol{X}_1 = \begin{bmatrix} 0 & 5.49 & 5.49 & 1.00 & 1.00 & 1.00 \\ 5.49 & 0 & 10.99 & 5.49 & 5.49 & 5.49 \\ 5.49 & 10.99 & 0 & 5.49 & 5.49 & 5.49 \\ 1.00 & 5.49 & 5.49 & 0 & 1.00 & 1.00 \\ 1.00 & 5.49 & 5.49 & 1.00 & 0 & 1.00 \\ 1.00 & 5.49 & 5.49 & 1.00 & 1.00 & 0 \end{bmatrix}.$$

Note that the effect of the outlier edge has dissipated, and the resulting metric graph does not have very large edge weights. On the other hand, the outlier edge leads to a significant change in edge weights when the Euclidean divergence is used:

$$\boldsymbol{X}_2 = \begin{bmatrix} 0 & 1667.33 & 1667.33 & 1.00 & 1.00 & 1.00 \\ 1667.33 & 0 & 3334.67 & 1667.33 & 1667.33 & 1667.33 \\ 1667.33 & 3334.67 & 0 & 1667.33 & 1667.33 & 1667.33 \\ 1.00 & 1667.33 & 1667.33 & 0 & 1.00 & 1.00 \\ 1.00 & 1667.33 & 1667.33 & 1.00 & 0 & 1.00 \\ 1.00 & 1667.33 & 1667.33 & 1.00 & 1.00 & 0 \end{bmatrix}.$$

**7.2. Learning a kernel matrix.** In data mining and machine learning applications, linear separators or hyperplanes are often used to cluster or classify data. However, linear separators are inadequate when the data is not linearly separable. To overcome this problem, the data can first be mapped (nonlinearly) to a higher-dimensional feature space, after which linear separators can be used in the transformed feature space.

Suppose the data belong to the set $\Omega$, and $f : \Omega \rightarrow \mathscr{X}$ maps the data to an inner-product space $\mathscr{X}$, called the [illegible]. Given data objects $\{u_1, u_2, \ldots, u_N\} \subset \Omega$, the Gram matrix $\boldsymbol{X}$ is the $N \times N$ matrix of inner products in the feature space, $x_{mn} = \langle f(u_m), f(u_n) \rangle = g(u_m, u_n)$. This Gram matrix is also called the [illegible], and it captures the similarity between the objects $u_m$ and $u_n$. When the data space $\Omega$ is an inner-product space, common kernels include the polynomial kernel $g(\boldsymbol{u}_m, \boldsymbol{u}_n) = \langle \boldsymbol{u}_m, \boldsymbol{u}_n \rangle^d$ and the Gaussian kernel $g(\boldsymbol{u}_m, \boldsymbol{u}_n) = \exp\left\{-\frac{1}{2}\|\boldsymbol{u}_m - \boldsymbol{u}_n\|_2^2/\sigma^2\right\}$. These kernels are both positive definite. Conversely, any positive-definite matrix can be thought of as a kernel matrix [38]. In general, the set $\Omega$ can be arbitrary. For example, $\Omega$ might contain nucleotide sequences of varying lengths or phylogenetic trees or arbitrary graphs.

In many such situations, the choice of the kernel matrix is unclear. There is often an approximate kernel matrix $\boldsymbol{Y}_0$ that we wish to modify based on our information about the underlying data objects. This information may take various forms:

- known values for kernel entries ($x_{mn} = \alpha$),
- known distances between objects in the feature space ($x_{mm}+x_{nn}-2x_{mn} = \beta$), or
- known bounds on kernel entries ($x_{mn} \leq x_{rs}$) or distances ($x_{mm} + x_{nn} - 2x_{mn} \leq \gamma$).

Such constraints are typically obtained from the application domain, such as information about whether a pair of genes or proteins is functionally more similar than another pair.

Suppose that we are given an approximate kernel matrix $\boldsymbol{Y}_0$. Our problem is to find the nearest positive-definite matrix to $\boldsymbol{Y}_0$ that satisfies linear equality and inequality constraints. The von Neumann divergence can be used as the nearness measure:

$$D_{\mathrm{vN}}(\boldsymbol{X};\boldsymbol{Y}) = \operatorname{Tr}\left[\boldsymbol{X}(\log \boldsymbol{X} - \log \boldsymbol{Y}) - \boldsymbol{X} + \boldsymbol{Y}\right].$$

Using the von Neumann divergence appears to be advantageous when the initial kernel matrix $\boldsymbol{Y}_0$ is of low rank and it is desired that its null space be preserved [25]. Recall that, in the low-rank case, the von Neumann divergence $D_{\mathrm{vN}}(\boldsymbol{X};\boldsymbol{Y}_0)$ is finite only when the null space of $\boldsymbol{X}$ contains the null space of $\boldsymbol{Y}_0$. Hence, both the null space constraint and positive semidefiniteness are automatically enforced by the successive projection–correction algorithm.

## 8. Open problems and conclusions.

The Bregman nearness problem is relatively unstudied, so it opens a rich vein of new questions. Here are some specific challenges that deserve attention.

1. The matrix divergences described in subsection 2.6 offer an intriguing way to compute distances between Hermitian matrices. It would be valuable to characterize different types of projections onto important sets of matrices, such as the positive-semidefinite cone, the nonnegative cone, or the set of diagonal matrices. This could lead to more efficient numerical methods for key problems.
2. The algorithms described in this paper apply only to projections onto polyhedral convex sets. Some important constraint sets—such as the positive-semidefinite cone—are not so simple. In this work, we avoided trouble by incorporating the positive-semidefinite constraint into the divergence, but this approach is not always warranted. For more general problems, a different approach is necessary.
3. A more serious problem with the successive projection approach is that it offers only linear convergence. For applications, it may be critical to develop algorithms with superlinear convergence.
4. The matrix functions that arise from the study of matrix divergences lead to another challenge. We are not aware of a sophisticated approach to calculating a function such as $\exp(\log \boldsymbol{Y} + \boldsymbol{A})$ other than to work with the corresponding eigendecompositions. Expressions of this form frequently arise in Bregman nearness problems, and we would like to have more robust,

efficient techniques for their computation. Moreover, the numerical stability of various techniques needs to be studied.

5. In applications, it is most important to determine what divergence is appropriate. This choice is likely to depend on domain expertise, coupled with a nuanced understanding of the properties of different divergences.
6. One can also imagine the problem of [illegible] a divergence from data. This method would be the ultimate way to match the distance measure with the application. The connection between divergences and exponential families even provides a theoretical justification for this approach.

In conclusion, we have offered evidence that Bregman divergences provide a powerful way to measure the distance between matrices. They can react to structure in the matrix in a way that the Frobenius norm does not. This property makes them extremely valuable for applications, although it may take some effort to determine what divergence is appropriate. Moreover, the numerical methods for computing Bregman projections are still in their infancy. These challenges must be faced before divergences can occupy their potential role in data analysis.

**Acknowledgments.** We would like to thank Nick Higham and two anonymous referees for a thorough reading and helpful suggestions.

## REFERENCES

[1] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha, *A generalized maximum entropy approach to Bregman co-clustering and matrix approximation*, J. Mach. Learn. Res., 8 (2007), pp. 1919–1986. Available online at http://jmlr.csail.mit.edu/papers/volume8/banerjee07a/banerjee07a.pdf.

[2] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, *Clustering with Bregman divergences*, J. Mach. Learn. Res., 6 (2005), pp. 1705–1749.

[3] O. Barndorff-Nielsen, *Information and Exponential Families in Statistical Theory*, John Wiley, New York, 1978.

[4] H. H. Bauschke and J. M. Borwein, *Legendre functions and the method of random Bregman projections*, J. Convex Anal., 4 (1997), pp. 27–67.

[5] H. H. Bauschke and P. L. Combettes, *Iterating Bregman retractions*, SIAM J. Optim., 13 (2003), pp. 1159–1173.

[6] H. H. Bauschke and A. S. Lewis, *Dykstra's algorithm with Bregman projections: A convergence proof*, Optimization, 48 (2000), pp. 409–427.

[7] S. Boyd and L. Xiao, *Least-squares covariance matrix adjustment*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 532–546.

[8] L. M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 200–217.

[9] L. M. Bregman, Y. Censor, and S. Reich, *Dykstra's algorithm as the nonlinear extension of Bregman's optimization method*, J. Convex Anal., 6 (1999), pp. 319–333.

[10] Y. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*, Numer. Math. Sci. Comput., Oxford University Press, Oxford, UK, 1997.

[11] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley, New York, 1991.

[12] I. Csiszár, *I-divergence geometry of probability distributions and minimization problems*, Ann. Probab., 3 (1975), pp. 146–158.

[13] W. E. Deming and F. F. Stephan, *On a least squares adjustment of a sampled frequency table when the expected marginal totals are known*, Ann. Math. Statist., 11 (1943), pp. 427–444.

[14] F. Deutsch, *Best Approximation in Inner Product Spaces*, Springer-Verlag, New York, 2001.

[15] I. S. Dhillon, S. Sra, and J. A. Tropp, *Triangle fixing algorithms for the metric nearness problem*, in Proceedings of the Eighteenth Annual Conference on Neural Information Processing Systems (NIPS), MIT Press, Cambridge, MA, 2005, pp. 361–368.

[16] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842.

[17] I. Grubišić and R. Pietersz, *Efficient rank reduction of correlation matrices*, Linear Algebra Appl., 422 (2007), pp. 629–653.

[18] S.-P. Han, *A successive projection method*, Math. Programming, 40 (1988), pp. 1–14.

[19] N. J. Higham, *Computing the nearest correlation matrix—a problem from finance*, IMA J. Numer. Anal., 22 (2002), pp. 329–343.

[20] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*, Springer, Berlin, 2001.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[22] C. T. Ireland and S. Kullback, *Contingency tables with given marginals*, Biometrika, 55 (1968), pp. 179–188.

[23] E. T. Jaynes, *Information theory and statistical mechanics*, Phys. Rev., 106 (1957), pp. 620–630.

[24] R. Kruithof, *Telefoonverkeersrekening*, De Ingenieur, 52 (1937), pp. E15–E25.

[25] B. Kulis, M. Sustik, and I. S. Dhillon, *Learning low-rank kernel matrices*, in Proceedings of the Twenty-Third International Conference on Machine Learning (ICML), Morgan Kaufmann, San Francisco, 2006, pp. 505–512.

[26] A. S. Lewis, *The convex analysis of unitarily invariant matrix functions*, J. Convex Anal., 2 (1995), pp. 173–183.

[27] A. S. Lewis, *Convex analysis on the Hermitian matrices*, SIAM J. Optim., 6 (1996), pp. 164–177.

[28] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 272–284.

[29] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*, Academic Press, London, 1979.

[30] M. V. Menon, *Reduction of a matrix with positive elements to a doubly stochastic matrix*, Proc. Amer. Math. Soc., (1967), pp. 244–247.

[31] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.

[32] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[33] B. N. Parlett and T. L. Landis, *Methods for scaling to doubly stochastic form*, Linear Algebra Appl., 48 (1982), pp. 53–79.

[34] H. Qi and D. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 360–385.

[35] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[36] R. Sinkhorn, *A relationship between arbitrary positive matrices and doubly stochastic matrices*, Ann. Math. Statist, 35 (1964), pp. 876–879.

[37] P. Tseng, *Dual coordinate ascent methods for non-strictly convex minimization*, Math. Programming, 59 (1993), pp. 231–247.

[38] V. N. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1998.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 4, pp. 1147–1170

# A GIVENS-WEIGHT REPRESENTATION FOR RANK STRUCTURED MATRICES*

STEVEN DELVAUX† AND MARC VAN BAREL†

**Abstract.** In this paper we introduce a Givens-weight representation for rank structured matrices, where the rank structure is defined by certain submatrices starting from the bottom left or upper right matrix corner being of low rank. We proceed in two steps. First, we introduce a unitary-weight representation. This representation will be compared to the (block) quasiseparable representations introduced by P. Dewilde and A.-J. van der Veen [*Time-varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998]. More specifically, we show that our unitary-weight representations are theoretically equivalent to the so-called block quasiseparable representations in input or output normal form introduced by Dewilde and van der Veen [*Time varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998]. Next, we move from the unitary-weight to the Givens-weight representation. We then provide some basic algorithms for the unitary/Givens-weight representation, showing how to obtain such a representation for a dense matrix by means of numerical approximation. We also show how to "swap" the representation and how to reduce the number of parameters of the representation, whenever appropriate. As such, these results will become the basis for algorithms on unitary/Givens-weight representations to be described in subsequent papers.

**Key words.** rank structured matrix, representation, elementary unitary operation, Givens transformation, numerical approximation

**AMS subject classifications.** 65F, 65F30, 15A03

**DOI.** 10.1137/060654967

## 1. Introduction.

In this paper we describe a way to obtain compact representations for rank structured matrices. More specifically, these will be the unitary-weight and Givens-weight representations, in increasing order of specification. The basic idea of our representations is a generalization from the so-called Givens-vector representation introduced in [17], which we generalize to the case of an arbitrary rank structure.

First, we must define the class of matrices for which our representations will be appropriate.

Definition 1 (see [3]). [illegible] $\mathcal{R}$ [illegible] $\mathbb{C}^{m\times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_k$ [illegible] $\mathcal{B}_k$ [illegible] 3 [illegible]

$$\mathcal{B}_k = (i_k, j_k, r_k),$$

---

[*] Received by the editors March 23, 2006; accepted for publication (in revised form) by N. Mastronardi April 20, 2007; published electronically November 21, 2007. This research was partially supported by the Research Council K.U. Leuven, project OT/05/40 (Large rank structured matrix computations), Center of Excellence: Optimization in Engineering, by the Fund for Scientific Research–Flanders (Belgium), G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), G.0423.05 (RAM: Rational modelling: optimal conditioning and stable algorithms), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The scientific responsibility rests with the authors.

http://www.siam.org/journals/simax/29-4/65496.html

[†] Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium (Steven.Delvaux@cs.kuleuven.ac.be, Marc.VanBarel@cs.kuleuven.ac.be).

[illegible] $i_k$ [illegible] $j_k$ [illegible] $r_k$ [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible] $\mathcal{R}$ [illegible] $k$

$$\mathrm{Rank} A(i_k : m, 1 : j_k) \leq r_k.$$

The above definition uses the word [illegible] to distinguish from the more general rank structures that were handled in [3]. Since these more general structures do not occur in the present paper, we will simplify notation by just dropping the word [illegible] everywhere from the notation.

Note that by definition, all structure blocks have to start from the lower left matrix corner. An example of a rank structure is shown in Figure 1.1.

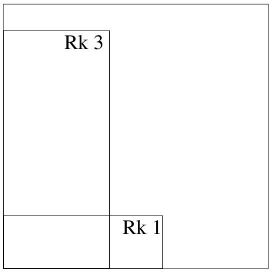


FIG. 1.1. *Example of a rank structure with two structure blocks $\mathcal{B}_1$ and $\mathcal{B}_2$. The notation "*Rk $r$*" denotes that the structure block is of rank at most $r$.*

In practice, it often happens that also the block [illegible] triangular part is rank structured, i.e., that also the matrix $A^T$ satisfies rank structure in the sense of Definition 1. By abuse of notation, we will indiscriminately use the term [illegible] also in this case.

It is easy to see that each structure block induces certain relations between the elements of the matrix. Hence it should be possible to represent a rank structured matrix using only a small number of parameters. Although one can devise several ways to obtain such a representation, there seem to be two classes of representations frequently used in the literature, which we call here $uv$ [illegible] and [illegible]. Let us give a brief survey.

The class of $uv$-representations was historically the first; see, e.g., [12]. We mention that this type of representation is possible only under certain conditions.

The different and more flexible class of block quasiseparable representations has been introduced in [7]. Starting from this book, many algorithms for these representations have been developed in the literature and appear in the work of Dewilde and van der Veen; see, e.g., [7, 8]. They were then used by Eidelman and Gohberg, who also introduced the name (block) "quasiseparable representation"; see, e.g., [9, 10, 11]. More recently, these matrices appear under the name of "sequentially semiseparable representations" in the work of Chandrasekaran, Gu et al.; see, e.g., [1]. Finally, we note that rank structured matrices appear also in a purely theoretical context in the work of Tyrtyshnikov [14], under the name of "weakly semiseparable matrices."

We are not intending to give here a complete overview of all the existing algorithms for block quasiseparable representations: this would be a task even more complicated by the sometimes varying conditions under which these algorithms are derived, most notably the fact that the underlying structure blocks must be situated just below the main diagonal, are equidistant, and so on, with the precise conditions depending sometimes from paper to paper. Instead, we will compare our own algorithms with those in the literature at the appropriate places in the remainder of this and following papers. We refer also to section 4 for a more detailed comparison with *uv*- and block quasiseparable representations.

For the special case of [illegible] matrices of semiseparability rank one, the latter defined by a collection of structure blocks $\mathcal{B}_k = (k, k, 1)$, $k = 1, \ldots, n$ on $\mathbb{C}^{n \times n}$, an alternative representation, the so-called [illegible], has been introduced in [17]. A first step in the generalization to higher semiseparability ranks was taken in [15]. In this paper, we will carry this scheme one step further, by generalizing the idea of the Givens-vector representation to be able to represent [illegible] rank structure.

Our generalization of [17] contains two layers of complication. First, we introduce the [illegible]. We show how these are theoretically equivalent to the so-called block quasiseparable representations in [illegible] or [illegible] introduced by Dewilde and van der Veen [7]. Next, we specify our representation to obtain the [illegible]. We will specify various types of such Givens-weight representations (e.g., those of type 1 and type 2), and we present several tools to deal with these representations in an efficient and stable way.

Briefly, it could be said that the present paper extends the earlier concept of the Givens-vector representation for semiseparable matrices of semiseparability rank one [17] to its natural framework, thereby significantly extending and illuminating these earlier concepts. For example, in our future work [4, 6] we use easily understandable operations like "extending and regressing the action radius" (the latter being a concept introduced in the present paper) instead of the seemingly ad hoc formulas used in [17, 16]. These methods are both stable and work under the condition of general rank structures.

One feature of our representations is that they can be used to obtain the (asymptotically) minimal number of parameters to represent the rank structured matrix part. More precisely, we show that (i) the Givens-weight representation can lead to a representation consisting of $O(rn)$ parameters, where $n$ is the matrix size and $r$ is a measure for the average rank index of the rank structure, and this for [illegible] rank structure; (ii) the unitary-weight representation (or by the same means, the block quasiseparable representations) can lead to such an $O(rn)$ representation [illegible]. The latter observation doesn't seem to be well known in the literature, in the sense that, e.g., many papers use a representation consisting of $O(r^2 n)$ parameters; see, e.g., [9], as well as many other papers by these same authors.

This paper is organized as follows. Section 2 introduces the basic ideas of the unitary-weight and Givens-weight representations. Section 3 considers the operations of approximating and reducing the representation. Section 4 contains a detailed comparison of the unitary/Givens-weight representation with the other kinds of representations in the literature. Finally, some conclusions are provided in section 5.

## 2. Givens-weight representation.

In this section, we will describe the basic ideas enabling one to obtain a compact representation for rank structured matrices. The representation will generalize the Givens-vector representation for semiseparable matrices of semiseparability rank one, which was introduced in [17].

The Givens-weight representation will be an [illegible] representation, which works strictly inside the area spanned by the structure blocks and considers the "outside world" to be inaccessible.

This section is organized as follows. In the first two subsections, we introduce the concepts of unitary-weight and Givens-weight representations, first on a special case in subsection 2.1, and later for general rank structures in subsection 2.2. In the last two subsections, we further complete the description of the representation, by specifying two directions in which it can be extended: we describe representations that are based on column instead of row operations in subsection 2.3, and for the upper instead of the lower triangular matrix part in subsection 2.4.

**2.1. Example.** In the present subsection we will try to indicate the underlying ideas of unitary-weight representations. To this end we will take the structure in Figure 2.1 as a didactical example. First, it may be noted that this figure does not show the surrounding matrix box anymore: this reflects a fact mentioned before, namely that only the area spanned by the structure blocks will be relevant for the representation and that the "outside world" will be inaccessible.

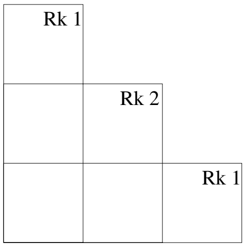


FIG. 2.1. *Example of a rank structure with three structure blocks $\mathcal{B}_1, \mathcal{B}_2$, and $\mathcal{B}_3$. We will use this example to explain the mechanism of the unitary-weight and Givens-weight representation during the following paragraphs. From now on the surrounding matrix box, as in Figure* 1.1*, will not be shown anymore.*

In what follows, we will often work with [illegible]. These are defined as unitary matrices having a block diagonal form $U = I_a \oplus Q \oplus I_b$, where $I_a, I_b$ denote identity matrices of suitable sizes. When such a unitary operation $U$ acts on the rows of a given matrix, we will represent it in a pictorial way by a vertical line segment, placed on the position of the rows on which it acts. Sometimes we will actually denote it as a vertical [illegible], instead of a line segment, as an auxiliary means for visualizing the algorithm flow; see below.

The unitary-weight representation is obtained by reducing the structure blocks into blocks of zeros, by the use of elementary unitary row transformations. First, we apply an (elementary) unitary transformation to transform the bottom Rk 1 block into a block of zeros, with one row less; see Figure 2.2.

Note that this unitary transformation acts only on the columns on the left of the vertical line which is indicated in boldface in the figure, in the present case rows $1, 2, \ldots, 9$. We say that this vertical line borders the [illegible] of the unitary transformation. Thus the action radius of the current unitary transformation is equal to 9.

Having applied this operation, note that in columns 7, 8, and 9 that we have already reached the "top" of the structure. Therefore, this is now the right moment to consider the top elements of these columns, and to store them. These elements will be called [illegible], and they are visualized on a grey background in Figure 2.2.

From now on we consider columns 7, 8, and 9 as finished, and we restrict our

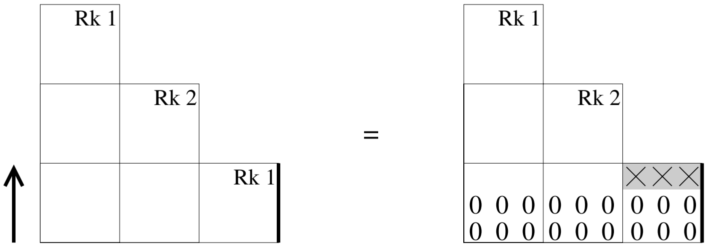


FIG. 2.2. *We apply a unitary transformation to transform the bottom two rows of the structure into zeros. This transformation acts only on the columns on the left of the vertical line which is indicated in boldface in the figure: this line borders the* action radius *of the unitary transformation. Having performed this unitary transformation, the elements indicated on a grey background are stored; they are called* weights.

perspective to the previous columns. We can then apply a unitary transformation to transform the middle Rk 2 block into a block of zeros, with two rows less; see Figure 2.3.

Note that again, this unitary operation acts only on the columns on the left of the vertical line indicated in boldface in the figure. Thus the action radius of the current unitary transformation is equal to 6.

Having applied this operation, note that also in columns 4, 5, and 6 we have reached the top of the structure. Therefore, this is now the right moment to consider the top elements of these columns and to store them. This yields us a second block of weights, which is again visualized on a grey background in Figure 2.3.

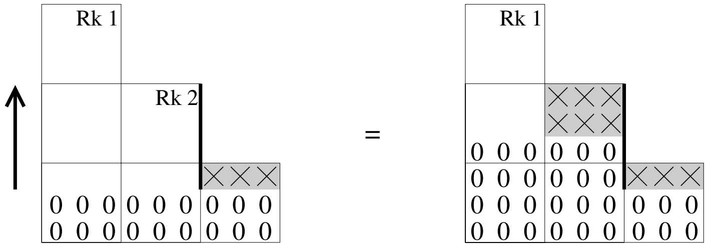


FIG. 2.3. *We apply the next unitary transformation and store the new block of weights.*

From now on we drop columns 4, 5, and 6 from our perspective. We can then apply a unitary transformation to transform the top Rk 1 block into a block of zeros, with one row less; see Figure 2.4. We conclude by storing the final block of weights.

The weights can now be collected into a single matrix, which we call the *weight matrix*. Together with the computed unitary transformations, this matrix yields us the complete *unitary-weight representation* of the given matrix; see Figure 2.5.

Of course, to be a useful representation, the unitary-weight representation should allow the possibility to restore the original matrix from which we started. This can be done by "reversing" the previous steps.

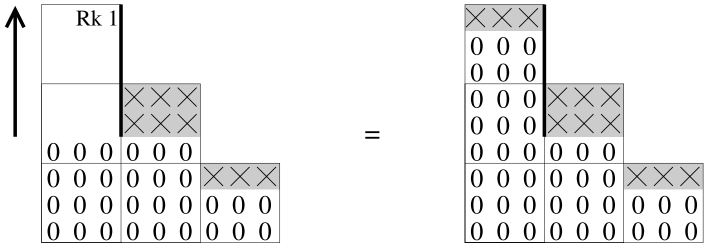


Fig. 2.4. *We apply the final unitary transformation and store the new block of weights.*

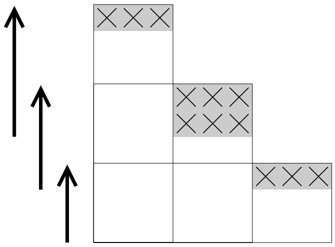


Fig. 2.5. *Schematic picture of the unitary-weight representation for the rank structure in Figure* 2.1.

This reversal process will be called [illegible] the unitary-weight representation and is shown in Figure 2.6. First, we start by spreading out the top block of weights, i.e., we multiply them with the inverse of the top unitary transformation of the unitary-weight representation; see Figure 2.6(a).

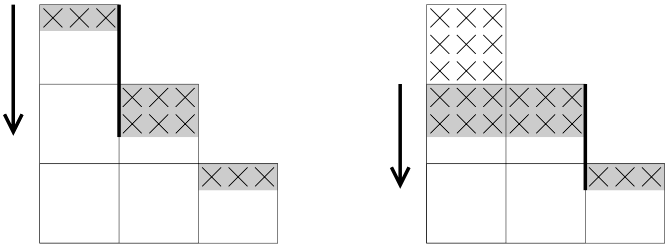
As a result of this spreading-out, the weight matrix will start to get filled-in. Moreover, since we are now at the point of entering the structure in columns 4, 5, and 6, this is now the right moment to also bring the middle block of weights into our perspective. Then we spread out all of these elements; see Figure 2.6(b).

As a result of this spreading-out, the weight matrix will get filled-in more and more. Moreover, since we are now at the point of entering the structure in columns 7, 8, and 9, this is now the right moment to also bring the last block of weights into our perspective. Then we spread out all of these elements; see Figure 2.6(c).

Finally, we then retrieve the original, full matrix which we started from; see Figure 2.6(d).

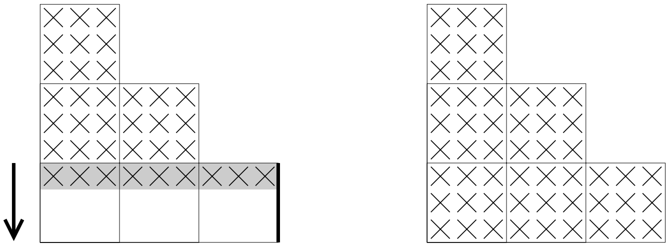
The reader will have noticed that we used a grey/white color code in Figure 2.6, as well as in the previous figures. Let us explain the meaning of this code. The grey elements are used to denote the [illegible], i.e., the elements that contain "condensed" information about the full matrix. Thus in order to see the real meaning of these elements, they first have to be spread out by the next unitary transformations of the unitary-weight representation. On the other hand, the white elements denote the [illegible], i.e., the elements that will not be further influenced by the next unitary transformations. These are actual elements of the full matrix.

This grey/white code turns out to be quite handy; therefore, it will be frequently used in what follows.



(a) Consider the first block of weights and spread out.

(b) Consider also the second block of weights and spread out.



(c) Consider also the third block of weights and spread out.

(d) We now retrieve the full matrix.

FIG. 2.6. *Spreading out the unitary-weight representation.*

**2.2. General definitions.** Now we are ready to handle the unitary-weight and Givens-weight representations in the general case. We will do this for a matrix $A$ satisfying some general rank structure $\mathcal{R} = \{\mathcal{B}_k\}$. Just as in the example of the previous subsection, we will have to assume that the structure blocks are ordered in a [illegible] way, i.e., that there is no pair of structure blocks for which the first one is completely contained in the second one. If this condition is not satisfied yet, then we first have to remove from the structure all such structure blocks which are completely contained in another structure block. (Actually, these nested structure blocks are not completely useless, in the sense that they lead to an additional sparsity pattern in the Givens-weight representation. But we will not be concerned about this here.)

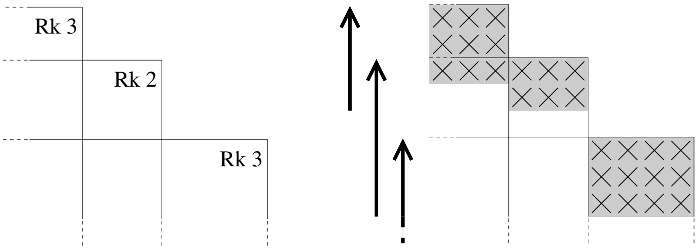
Thus we can now assume that the rank structure does not contain any nested structure blocks anymore. We can then order the remaining structure blocks in a sequential way, going from the top left to the bottom right corner of the matrix. Stated in another way, the structure blocks are ordered in such a way that both the row and column indices $i_k$ and $j_k$ of the structure blocks increase in a strictly monotonic way.

Then we can come to the general definition of unitary-weight representations.

DEFINITION 2 (index sets). [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_{k=1}^{K}$ [illegible] $i_1 < \cdots < i_K$ [illegible] $j_1 < \cdots < j_K$ [illegible]

[illegible] $I_k = \{i_k, \dots, i_{k+1} - 1\}$ $I_{k,\text{top}} = \{i_k, \dots, i_k + r_k - 1\}$ [illegible] $J_k = \{j_{k-1}+1, \dots, j_k\}$ [illegible] $k = 1, \dots, K$ [illegible] $i_{K+1} := N + 1$ [illegible] $j_0 := 0$ [illegible] $r_{K+1} := 0$

Definition 3 (unitary-weight representation). [illegible] $A \in \mathbb{C}^{m \times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}_{k=1}^{K}$ [illegible] $i_1 < \cdots < i_K$ [illegible] $j_1 < \cdots < j_K$ [illegible] unitary-weight representation [illegible] $A$ [illegible] $\mathcal{R}$ [illegible] $(\{U_k\}_{k=1}^{K}, W)$. [illegible] $U_k$ [illegible] $I_k \cup I_{k+1,\text{top}}$ [illegible] $\bigcup_{l=1}^{k} J_l$ [illegible] $I_{k,\text{top}}$ [illegible] $U_k$ [illegible] $k = K, \dots, 1$ [illegible] $W \in \mathbb{C}^{m \times n}$ [illegible] weight matrix [illegible] $I_{k,\text{top}}$ [illegible] $J_k$ [illegible] $U_k$ [illegible] 2.7



(a) (b)

Fig. 2.7. *For the rank structure in the left picture, the right figure shows a schematic picture of the unitary-weight representation.*

If a unitary-weight representation of a matrix is given, then we can restore the full matrix by [illegible] the representation, which we explained in the previous subsection.

Now we will add some extra constraints to the above definition of unitary-weight representation, by additionally splitting each unitary transformation $U_k$ into a product of Givens tranformations. This will lead us to the description of [illegible]

In what follows, we will use the term [illegible] to denote a unitary operation which differs from the identity matrix in only two subsequent rows $i$ and $i + 1$. This transformation will be sometimes denoted as $G_{i,i+1}$, and the index $i$ will be called the [illegible] of the Givens transformation. Similarly to our notation for elementary unitary operations, we will graphically denote the Givens transformation $G_{i,i+1}$ by means of a vertical line segment, with the height at which this line segment is standing in the figure determined by the row index $i$ (see below).

First, rather than individual Givens transformations, it will be useful to work with [illegible]: these are defined as products of the form $G_{i+k,i+k+1} \cdots G_{i,i+1}$, for some $k \geq 0$. Graphically, this can be considered as a collection of Givens transformations where each Givens transformation is situated precisely one position below the previous one; see Figure 2.8.

The number of Givens tranformations of which a Givens arrow consists will be called the [illegible] of the Givens arrow. Moreover, we define the [illegible] and the [illegible]

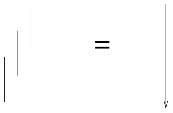


FIG. 2.8. *A Givens arrow* $G_{i+2,i+3}G_{i+1,i+2}G_{i,i+1}$ *consisting of three Givens transformations. Concerning this figure, we recall that we consider each Givens transformation as "acting" on the rows of an (invisible) matrix standing on the right of it, and hence that the Givens transformations in the figure should be evaluated from right to left, thereby explaining the downward direction of the Givens arrow.*

of the Givens arrow to be the largest and the smallest row index of the Givens transformations of which the Givens arrow consists, respectively. These notions have an obvious graphical interpretation.

Having introduced all of these notions, we can now move from unitary-weight to Givens-weight representations.

DEFINITION 4 (Givens-weight representation). [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible] $\mathcal{R} = \{\mathcal{B}_k\}$ [illegible] $i_1 < \cdots < i_K$ [illegible] $j_1 < \cdots < j_K$ [illegible] Givens-weight representation [illegible] $A$ [illegible] $\mathcal{R}$ [illegible] $U_k$ [illegible]

- [illegible] $r_k$
- [illegible] $U_k$ [illegible]

[illegible] 2.9

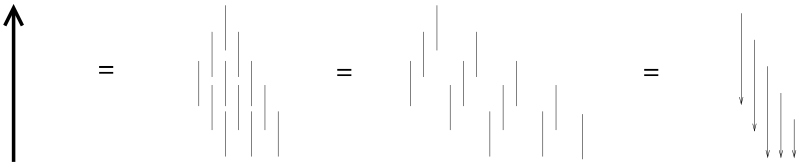


FIG. 2.9. *Suppose that the current structure block is* Rk 3 *and that the corresponding unitary transformation* $U_k$ *spans over six rows. Then we assume for this unitary transformation a decomposition into a product of Givens arrows of width at most* 3.

Let us comment on Figure 2.9. The left part of the figure denotes a unitary component $U_k$ of the Givens-weight representation. The middle and rightmost part of the figure then show the required decomposition of this unitary component $U_k$ into a product of Givens transformations. In particular, the equivalence between the two rightmost pictures in Figure 2.9 follows by repeatedly inserting Figure 2.8. On the other hand, the equivalence between the two [illegible] pictures in Figure 2.9 is purely aesthetic: these are two different ways for visualizing the same product of Givens transformations.

We should still explain why the assumption is made that each Givens arrow in the decomposition of $U_k$ has width at most $r_k$. To this end, recall that the unitary transformation $U_k$ serves to create zeros in a certain $\mathrm{Rk}(r_k)$ submatrix, except for its top $r_k$ rows. This effect can always be realized by a succession of Givens arrows as prescribed; see section 3 for more details.

It turns out that by decomposing each unitary transformation $U_k$ as specified in Definition 4, we formally obtain a decomposition into a product of [illegible] Givens transformations, in the sense that the beginning and trailing Givens transformations of two subsequent unitary transformations $U_k$ may overlap. This overlap may be especially severe when the structure is [illegible], i.e., when the vertical gaps between two subsequent structure blocks are small compared to their rank upper bounds; see Figure 2.11.

Let us comment on Figure 2.11. The figure shows a product of three subsequent unitary components $U_1U_2U_3$ of the Givens-weight representation, together with its decomposition as a product of Givens transformations. This leads to a total of $3 \times 6 = 18$ Givens transformations. However, it can be shown theoretically that the compression process for which $U_1$, $U_2$, and $U_3$ are intended can be performed using only 12 Givens transformations. More precisely, the 6 Givens transformations which are encircled in Figure 2.11 are, in principle, redundant for this compression process.

In practical situations we would like to avoid such a superfluous amount of Givens transformations. We will now describe some extreme cases where this is achieved.

Definition 5 (Givens-weight representations of type 1 and 2). [illegible] 4 [illegible] $U_k$ [illegible] of type 1 [illegible] ” [illegible] ” [illegible] of type 2[1]

It turns out that [illegible] Givens-weight representation can be brought into each of the forms of the above definition. This reduction can be achieved by means of so-called “pull-through techniques.” The following lemma is pivotal in this respect.

Lemma 6 (pull-through lemma). [illegible] 3 by 3 matrix $Q$ which is factorized as

$$Q = G'_{1,2}G_{2,3}G_{1,2},$$

then there exists a refactorization

$$Q = \tilde{G}'_{2,3}\tilde{G}_{1,2}\tilde{G}_{2,3}.$$

See Figure 2.10.

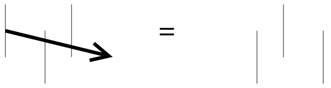


Fig. 2.10. *Pull-through lemma applied in the downward direction. One could imagine that the leftmost Givens transformation is really “pulled through” the two rightmost Givens transformations, as indicated by the bold right downward pointing arrow.*

The above version of the pull-through lemma was formulated in the downward direction. In a similar way there exists a pull-through lemma in the [illegible] direction, but we will not be concerned about this here.

---

[1] We may point out that under some additional conditions, the Givens-weight representations of type 1 and 2 have an interpretation in terms of “zero-creating” and “rank-decreasing” Givens patterns, respectively, as discussed in [6]. We should mention that the zero-creating Givens pattern has also been discussed in [5], although it was done there in a more theoretical context.

As an example, let us repeatedly apply the pull-through lemma in order to reduce the number of parameters of the representation in Figure 2.11. The resulting situation is then shown in Figure 2.12. Note that the superfluous Givens transformations have been removed by the pull-through process, and that the representation has become of type 1. Also the representation of type 2 can be obtained in this way, namely by repeatedly applying the pull-through lemma in the [illegible] direction.

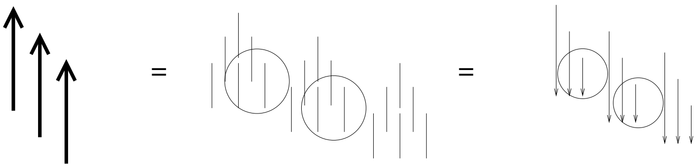


FIG. 2.11. *The leftmost picture shows a product of three subsequent unitary components* $U_1U_2U_3$ *in case of a dense rank structure* $\mathcal{R}$*. The two rightmost pictures then show the decomposition of this unitary matrix as a product of individual Givens transformations. The redundant Givens transformations are encircled.*

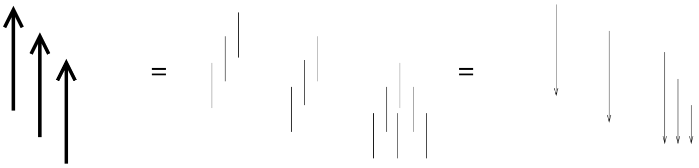


FIG. 2.12. *After removal of the superfluous Givens transformations in Figure* 2.11 *by, for example, repeatedly applying the pull-through lemma, we end up with a Givens-weight representation of type* 1.

Note that for efficiency reasons, the pull-through operations should be grouped in such a way that we move only one time from the top to bottom of the matrix. Moreover, each time we should "enlarge the action radius" of the Givens transformations which will be pulled-through, in order to give them the same action radius as the next unitary operation $U_k$; we refer to [6] for details.

In a certain sense, the representations of Definition 5 represent two extreme cases having a small amount of Givens transformations. For practical computations we can afford to work with a more general class of Givens-weight representations, which we call [illegible].

DEFINITION 7 (efficient Givens-weight representation). [illegible] 4 [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible] $\mathcal{R}$ [illegible] efficient [illegible] 1 [illegible] $\mathcal{R}$ [illegible] $m \approx n$ [illegible] $rn$ [illegible] $r$ [illegible]

This definition says that a Givens-weight representation can be efficient only if the number of superfluous Givens transformations is not too high w.r.t. the "optimal" value. As an illustrative example, the reader could keep in mind the difference between Figures 2.11 and 2.12.

Summarizing, we have now completely described the Givens-weight representation induced by a rank structure $\mathcal{R}$ and some of its practical variants. For the remainder of this section, we will indicate how this entire terminology can be reformulated in two other situations.

**2.3. Row versus column operations: swapping the representation.** The unitary-weight and Givens-weight representations described previously have the feature that the elementary unitary operations $U_k$ are considered as [illegible] operations. In a completely analogous way, one could build a representation based on [illegible] operations.

We are not intending to reproduce all of the previous definitions and examples to the column case. Instead, we will focus here on an efficient [illegible] to change from a row to a column representation or vice versa, hereby generalizing the swapping algorithm for semiseparable matrices of semiseparability rank 1 introduced in [16].

The swapping algorithm will be fairly simple and, together with its generalization to "generalized swapping" described in [4, 6], it will form a key ingredient for manipulating Givens-weight representations. Moreover, in section 3 we will come to a possible application of the swapping algorithm by showing that it can be used to perform an additional compression of the representation, i.e., to approximate the matrix by one having smaller ranks in its underlying rank structure, whenever appropriate.

The swapping process is illustrated in Figure 2.13.

Let us comment on this figure. Figure 2.13(a) shows the initial weight matrix, in which we have started to spread out the Givens transformations belonging to the first unitary transformation. We would then like to go on by also spreading out the next unitary operations, so that we can finally obtain the full version of the given rank structured matrix.

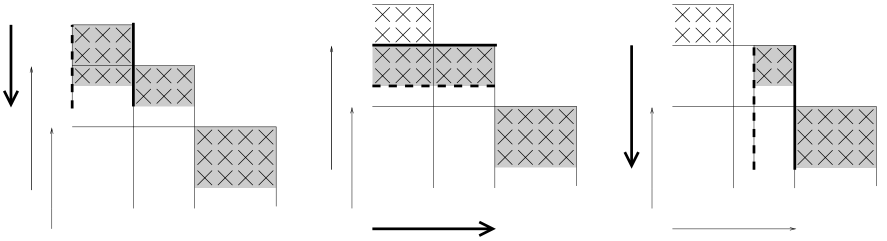
Before spreading out further, however, we apply an auxiliary column operation in order to bring the weights as much as possible to the right; see Figure 2.13(b). This auxiliary column operation serves to prevent the matrix from getting completely filled-in by the spreading-out process. It is applied only to the rows lying between the thick horizontal lines shown in Figure 2.13(b); in the present case rows $3, 4$.

Having done this, we can now go on to further spread out the rank structured matrix, while each time applying a unitary column operation to bring the weights as much as possible to the right. Figures 2.13(c), 2.13(d), 2.13(e), and 2.13(f) show the next steps in this swapping process.

In these figures, we used the following graphical code. The "active" unitary operation which is currently being applied to the rows or columns of the matrix is always shown in boldface. This may be both a unitary operation belonging already to the unitary-weight representation (for those acting on the rows), or a new unitary operation coming from outside (for those acting on the columns). On the other hand, unitary operations belonging to the unitary-weight representation, but which are not active in the current step of the algorithm, are always shown by [illegible] arrows.

The final situation in Figure 2.13(f) shows how we have completely spread out the weight matrix by annihilating the action of the original row operations, and where at the same time the weight matrix has again been compressed by the use of auxiliary column operations. In other words, we have now completely switched from a row-based to a column-based unitary-weight representation.

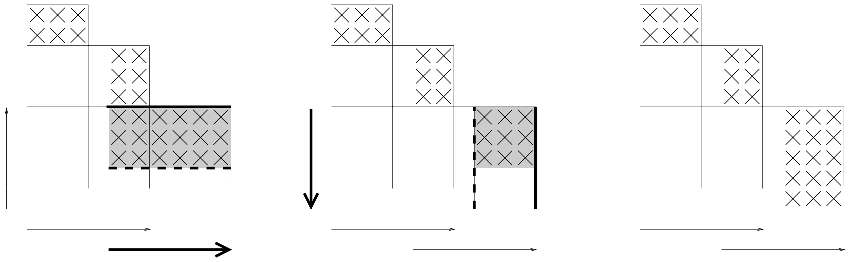
Note that also the weight blocks of the column representation can easily be read off during the swapping algorithm.



(a) Spread out the first unitary row operation.

(b) Apply a unitary column operation to bring the weights to the right.

(c) Spread out the second unitary row operation.



(d) Apply a unitary column operation to bring the weights to the right.

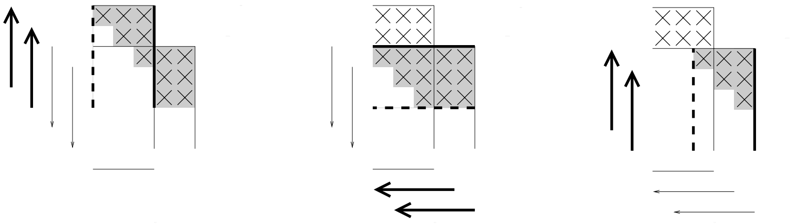
(e) Spread out the third unitary row operation.

(f) We now obtain the column-based Givens-weight representation.

FIG. 2.13. *Swapping the unitary-weight representation.*

For this algorithm, as well as for many other algorithms to be described, the algorithm was illustrated for a unitary-weight rather than a Givens-weight representation. But this is only for clarity reasons; in reality the above algorithm is capable of both exploiting and preserving the sparsity pattern satisfied by the Givens transformations. The main observation for this purpose is that by definition, the Givens transformations are organized in [illegible], each of them pointing in the opposite direction w.r.t. the algorithm flow; see Definition 4. This will guarantee that there is no superfluous fill-in during the algorithm in the sense that, loosely speaking, each Givens transformation will cause only [illegible] weight element to get filled in. The latter fact will then guarantee that the number of swapped Givens transformations is of the same order as the number of original Givens transformations; see Figure 2.14.

Let us consider the numerical complexity of the swapping algorithm. To this end we can first note that each Givens transformation on both rows and columns acts on a number of approximately $r$ weight elements, where $r$ is a measure for the average semiseparability rank of the rank structure $\mathcal{R}$. In particular, if the Givens-weight representation is efficient in the sense of Definition 7, and assuming that we work with a practical choice of rank structure, it follows that the complexity reduces to $O(r^2n)$.



(a) Starting situation. The top weight block, resulting from earlier swapping operations, will now be spread out by means of the inverses of the next Givens arrows (to be evaluated from left to right).

(b) We apply two Givens arrows (to be evaluated from top to bottom) to the columns to bring the weights to the right.

(c) We could now go on to spread out by means of the inverses of the next Givens arrows (to be evaluated from left to right) on the rows, and so on.

Fig. 2.14. *The figure shows a more detailed illustration of the swapping operation, in terms of individual Givens arrows. Note that the sparsity of the weight block in Figure* 2.14(a) *has been preserved during the algorithm, in the sense that the weight block in Figure* 2.14(c) *satisfies the same sparsity. Moreover, this sparsity allows the number of swapped Givens transformations determined in Figure* 2.14(b) *to be of the same order as the number of original Givens transformations.*

**2.4. Representation for the upper triangular part.** We will now focus on a second way to extend the definition of Givens-weight representation. Until now we have considered only rank structures in the [illegible] triangular part. But it is not difficult to generalize these notions to a Givens-weight representation for the structured [illegible] triangular part. Formally speaking, we define a rank structure in the block upper triangular part[2] as a collection $\mathcal{R}^T$ of structure blocks satisfied by the transpose matrix $A^T$. It is then straightforward to generalize the notion of Givens-weight representation to this case.

Note that if the matrix satisfies rank structure in both its block lower and block upper triangular part, we will obtain in this way a representation for the [illegible] matrix, as we describe now.

Definition 8. [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible]

- [illegible] $A$
- [illegible] $A$ [illegible]
- [illegible]

[illegible] Givens-weight representation [illegible] $A$ [illegible]

---

[2] The precise meaning of the term "block upper triangular part" or its synonym "structured upper triangular part" depends of course on the underlying distribution of the structure blocks. In particular, the structure blocks are allowed to go beyond the main diagonal. The same remarks hold of course for the block *lower* triangular part.

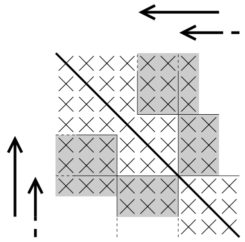


FIG. 2.15. *Schematic picture of a Givens-weight representation for a matrix A having rank structure both in its structured lower and in its structured upper triangular part. The figure shows the weight matrix as well as the unitary transformations of the representation. Note that the matrix is assumed here to be symmetric, and that the lower and upper triangular representations are based on row and column operations, respectively.*

[illegible] weight matrix [illegible] 2.15

Note that both the upper and the lower triangular part could be represented by either a row-based or a column-based Givens-weight representation, or even "hybrid" versions of this, leading to a total of at least $2\times 2 = 4$ different ways of representing the matrix $A$. The example in Figure 2.15 shows just one example of such an assignment.

Summarizing, we have now described the basic concepts of unitary/Givens-weight representations for rank structured matrices. In the remainder of this paper, as well as in the papers [4, 6], we will move to the development of practical algorithms for this representation.

## 3. Approximating and building the representation.

In this section we show how a unitary/Givens-weight representation can be built to approximate a dense matrix, and how a unitary/Givens-weight representation can be approximated to have lower ranks in its underlying rank structure. It will suffice to describe these operations for a representation of the block lower triangular part; the block upper triangular part can then be handled in a similar way.

This section is organized as follows. Subsections 3.1 and 3.2 describe how to approximate a dense matrix by a rank structured one, and how to approximate a rank structured matrix by one with smaller ranks, whenever appropriate. Subsection 3.3 discusses the complexity of the algorithms. Some comparisons with the literature are provided in subsection 3.4.

### 3.1. Approximating a dense matrix.

We start by building a unitary/Givens-weight representation to approximate the block lower triangular part of a dense matrix. The basic ideas of this process have already been given in section 2, where it was explained how the representation requires at the $k$th step the determination of an elementary unitary transformation $U_k$ that compresses a certain $\mathrm{Rk}(r_k)$ matrix, except for its top $r_k$ rows, $k = K, \ldots, 2, 1$. Moreover, it was explained there how the concept of Givens-weight representation requires this unitary operation $U_k$ to be decomposed in a certain way into a product of Givens arrows of width at most $r_k$; see Figure 2.9.

To achieve this in a practical way, we will first consider the case where the required $\mathrm{Rk}(r_k)$ matrix has been decomposed in the form of a truncated singular value decomposition, or more generally any rank-revealing factorization $GH^H$ with $G$ and $H$ both having precisely $r_k$ columns. The required unitary transformation $U_k$ can then be determined by simply applying Givens arrows to perform a QR-factorization of the generator $G$; in this way the decomposition into Givens arrows will have precisely the form required in Figure 2.9.

We may mention that this scheme can be improved to obtain a more compact form for the Givens arrows. To this end we first apply an auxiliary unitary [illegible] operation $C \in \mathbb{C}^{r_k \times r_k}$ to the generator $G$ in order to bring its bottom square submatrix into upper triangular form. Having done this, the number of Givens arrows required to perform the QR-factorization of the new generator $GC$ can be considerably lower. Having performed this QR-factorization, we can again remove the influence of the auxiliary column operation $C$ by multiplying with its inverse to the columns; the resulting generator $G$ will then still be zero except for its top $r_k$ rows, as required. We could then go on with the next structure block.

It can be shown that with a proper implementation of this scheme, each of the applied Givens arrows will [illegible] eliminate a row, which will then not be touched anymore by any of the following operations. In other words, this means that the heads of the subsequent Givens arrows will be strictly monotonically proceeding upwards, and hence that we will have obtained a Givens-weight representation of type 2.

Consider now the case where the given $\mathrm{Rk}(r_k)$ matrix is compressed by means of a rank-revealing QR-factorization [13]. We recall that this procedure consists of searching for the column with largest norm and bringing it in upper triangular form. The latter operation can be achieved, e.g., by means of a sequence of Givens transformations constituting an upward pointing Givens arrow. We can then remove the top row from our perspective and apply the same procedure to the remaining matrix. This procedure is then repeated until the remaining matrix is numerically zero, which will be the case after at most $r_k$ steps. The remaining numerically zero elements are simply truncated to zero.

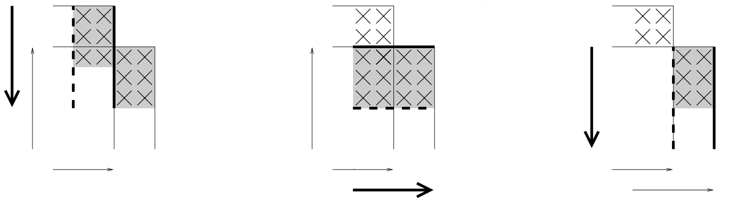
Now it is straightforward to see that the truncated rank-revealing QR-factorization will lead to a compression of the given $\mathrm{Rk}(r_k)$ matrix except for its top $r_k$ rows. The required elementary unitary transformation $U_k$ achieving this is simply the product of the applied [illegible] pointing Givens arrows of the rank-revealing QR-factorization; more precisely, it is obtained by rearranging these Givens transformations into [illegible] pointing Givens arrows of width $r_k$; see the second transition in Figure 2.9.

Just as in the previous paragraphs, it is possible to improve this scheme in order to obtain a more compact representation. This follows by remarking that the rank-revealing QR-factorization induces in a natural way a factorization $GH^H$, where $H^H$ contains the top $r_k$ rows of the compressed $\mathrm{Rk}(r_k)$ matrix, and the matrix $G$ contains the first $r_k$ columns of the (inverse of the) applied elementary unitary operation $U_k$. We can then proceed as described in the previous paragraphs to obtain a Givens-weight representation of type 2.

[illegible] 9.

1. The above discussion assumes that the exact ranks of the full matrix were known. Nevertheless, all the tools that we describe allow one to determine these ranks in a [illegible] way as the algorithm proceeds, depending on some numerical error threshold $\epsilon$. Hence we can really [illegible] the given matrix by a rank structured one.
2. Clearly, one can apply similar techniques to obtain a unitary-weight instead of a Givens-weight representation.

**3.2. Approximating a compressed matrix.** We will now show how to approximate a matrix with available unitary/Givens-weight representation by one with



(a) Starting situation. The top weight block, resulting from earlier swapping operations, will now be spread out by means of the inverse of the next unitary row operation $U_k$.

(b) Apply a unitary operation to compress the weights. Since the numerical rank is equal to two, we can compress until there are only two numerically nonzero columns left.

(c) We could now go on to spread out by means of the inverse of the next unitary row operation $U_{k+1}$, and so on.

FIG. 3.1. *Specification of Figure* 2.14 *in case the numerical ranks of the structure blocks are equal to two. We will then be able to additionally compress the representation.*

lower ranks in its underlying rank structure. Thus let us assume that the unitary/Givens-weight representation has led to an [illegible] of the numerical ranks of the underlying structure blocks, at least up to some numerical error threshold $\epsilon$. This fact will then be revealed during the swapping process, where now an additional compressing of the weight matrix can be performed; see Figure 3.1.

As can be seen, the above reduction algorithm follows by an easy application of the swapping procedure.

Let us describe here yet another closely related application of the swapping procedure, namely, the process of going from a "coarse" to a "fine" rank structure, in the sense of Figure 3.2. The corresponding algorithm is shown in Figure 3.3.

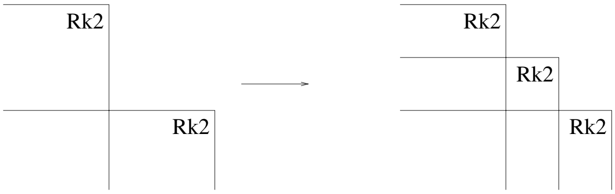
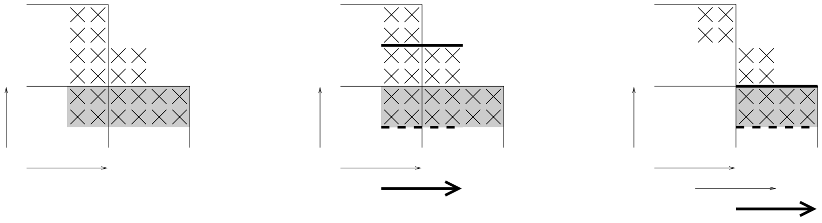


FIG. 3.2. *Going from a coarse to a fine rank structure. It is assumed here that the "intermediate" structure block is known to be numerically of rank at most two.*

It is easy to see that the process in Figure 3.3 can be extended to deal with [illegible] intermediate structure blocks, at least when these intermediate structure blocks are treated from top left to bottom right, corresponding to the flow direction of the swapping process.

**3.3. Complexity issues.** We will now describe some complexity issues for the approximation algorithms described in the previous subsections. First, note that the illustrations in Figures 3.1 and 3.3 have been expressed in terms of a unitary-weight instead of a Givens-weight representation. We already mentioned that this is done



(a) Starting situation. We assume that the swapping process has been performed up to the indicated unitary row operation $U_k$.

(b) Compress the indicated submatrix, which is known to be numerically of rank at most two, by an auxiliary column operation.

(c) We could now go on with the swapping process, by applying the next auxiliary column operation. After that, we spread out by means of the inverse of the row operation $U_k$, and so on.

FIG. 3.3. *Going from a coarse to a fine rank structure in terms of the unitary/Givens-weight representation, for the example in Figure* 3.2. *The algorithm performs a swapping procedure, during which the intermediate structure blocks are absorbed into the structure. Although the figure shows the case of a* single *intermediate structure block, the situation is similar to the case of* several *intermediate structure blocks.*

mainly for clarity reasons, but in the present case there is also a more fundamental reason, as we explain now.

The reader should recall that for the swapping algorithm of section 2, the particular sparsity pattern of the Givens-weight representation allowed for an additional benefit, in the sense that the swapped Givens arrows could be immediately read off in terms of the original Givens transformations. This fact was reflected by the Hessenberg-like shapes of the grey weight matrices in Figure 2.14.

In contrast, these observations are [illegible] true anymore for the algorithms in the previous subsections. The reason is that we are dealing here with [illegible] algorithms, and that the additional compression of the weight blocks should be performed by means of a numerically stable method such as a truncated singular value decomposition, a pivoted QR-factorization, or a similar routine. But to the best of our knowledge, such routines are unable to exploit the given shape of the matrix, in the sense that, denoting with $r$ a number such that the number of rows, the number of columns, the original ranks, and the numerical ranks are all of order $r$, then the complexity is $O(r^3)$, even if the original matrix was given in a Hessenberg-like form.

Of course, one can always use the techniques described in subsection 3.1 (but now with the role of rows and columns reversed) in order to obtain a reduced representation of type 2 (in the sense of Definition 5).[3] Alternatively, the superfluous Givens transformations could be removed afterwards by means of the pull-through lemma. But the point that we want to make here is that these techniques cannot improve on the $O(r^3)$ complexity per step. Thus in the case of a [illegible] rank structure, i.e., when the gaps between the subsequent structure blocks are very small, the reduction

---

[3] It suffices to apply each time an auxiliary row operation to bring the left square submatrix of the generator $H^H$ in upper traingular form, and only then determine the required Givens arrows on the columns as explained in subsection 3.1.

process will be of total complexity $O(r^3 n)$, which is rather inefficient.

A solution may be to work with only $O(\frac{n}{r})$ structure blocks, i.e., to choose the gaps between the structure blocks to be of order $r$. At the rate of an $O(r^3)$ complexity per step, the total complexity reduces then to $O(r^2 n)$. This type of solution was also sometimes observed in the literature; see, e.g., [2, page 13].

Finally, we recall once more that the complexity problems described previously are really inherent to the [illegible] problems occurring in the current section. In contrast, most other algorithms to be described in this and further papers [4, 6] will be [illegible] and therefore able to both exploit and preserve the sparsity of the Givens-weight representation, disregarding the underlying distribution of structure blocks.

**3.4. Comparison to the literature.** We close this section with some references to similar algorithms in the literature. Approximation algorithms were already given in [7, Chapters 3 and 5] in a more general operator-theoretical context. In particular, algorithms were provided there to obtain a block quasiseparable representation in input or output normal form, which are the theoretical equivalents of unitary-weight (but not Givens-weight) representations; see section 5. A more recent reference about approximation algorithms can be found in [2, section 11]. See also [11].

**4. Comparison with other representations.** In this section, the unitary/Givens-weight representation will be compared with two other representations frequently encountered in the literature: block quasiseparable representations and $uv$-representations. For both representations we describe an easy algorithm to transform the representation into the unitary/Givens-weight form. Moreover, it will be shown that especially the class of [illegible] representations is tightly connected to the unitary-weight format, and the similarities and differences between these two types of representation will be emphasized.

**4.1. Quasiseparable representations.** First, we will focus on block quasiseparable representations. The idea for these representations has essentially been introduced in the book [7]. In this paper, we will follow the terminology of Eidelman and Gohberg by calling these matrices block quasiseparable [9], but the reader should be aware that they also appear under other names in the literature, such as sequentially semiseparable matrices, matrices with low Hankel rank, and so on.

We will start with the most general definition of block quasiseparable matrices. A matrix is called [illegible] w.r.t. a given block partition if it can be divided as a block matrix w.r.t. this partition, with $(i, j)$th block element given by

$$\begin{cases} A_{i,j} = P_i T_{i-1} T_{i-2} \dots T_{j+1} Q_j, & i \geq j+1, \\ A_{i,j} = D_i, & i = j, \\ A_{i,j} = G_i S_{i+1} S_{i+2} \dots S_{j-1} H_j, & i \leq j-1 \end{cases}$$

for $i, j = 0, \dots, K$. Here an empty product denotes the identity matrix. The sizes of all auxiliary matrices occurring in these formulas must be chosen to be compatible with each other. In the literature, these matrices must sometimes satisfy some additional size restrictions, such as the fact that the $D_i$ are scalar (these lead then to the usual, i.e., scalar [illegible] matrices, occurring in many papers by Eidelman and Gohberg), the fact that the $D_i$ are square, and so on. These conditions vary sometimes from paper to paper.

We will refer to the matrices $T_k$ and $S_k$ as [illegible], although this terminology is somewhat nonstandard. Moreover, we will refer to the $P_i$, $G_i$ and $Q_j$, $H_j$ as [illegible], respectively.

The block partition corresponding to a given block quasiseparable representation immediately reveals the shape of the underlying structure blocks $\mathcal{B}_k$. In what follows, we will assume that these structure blocks $\mathcal{B}_k$, $k = 1, \ldots, K$ are labeled from top left to bottom right. Note also that the block quasiseparable formulae imply a certain limitation about the relative positions of the structure blocks in the block lower versus the block upper triangular part, but this restriction is not essential.

Figure 4.1 shows the block quasiseparable representation in a schematic way.

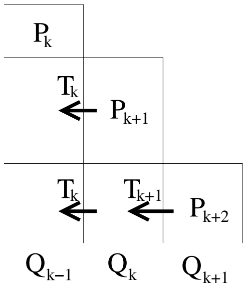


FIG. 4.1. *Schematic picture of a block quasiseparable representation. In order to obtain the* $(i, j)$*th block element, we should multiply the corresponding row shaft generator* $P_i$ *and the corresponding column shaft generator* $Q_j$*, with in between the product of all transition matrices* $T_k$ *that are needed to go from* $P_i$ *to* $Q_j$ *in the picture.*

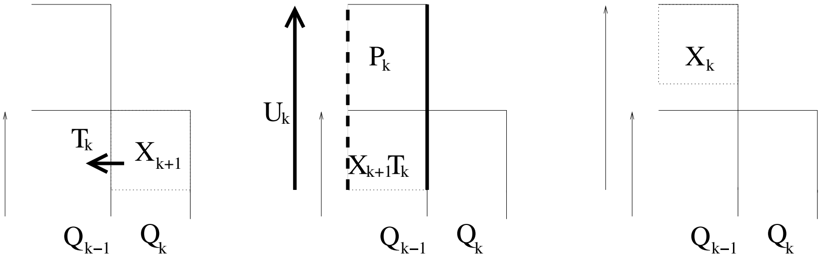
Now we consider the problem of transforming a given block quasiseparable representation into the Givens-weight format. We will restrict ourselves to the structured lower triangular part: the Givens transformations constituting the first unitary transformation $U_K$ can be derived from the QR-factorization of $P_K$. Then denoting by $X_K$ the square top part of the resulting R-factor, the Givens transformations constituting the next unitary operations $U_k$ can be derived from the QR-factorization of $\begin{bmatrix} P_k \\ X_{k+1}T_k \end{bmatrix}$, for $k = K-1, \ldots, 1$. Here we define each time the new $X_k$ to be the square top part of the resulting R-factor. During this process, the corresponding weight blocks can also be computed each time as $X_k Q_{k-1}$; see Figure 4.2.

Concerning the complexity of this algorithm, it is easy to see that the algorithm has a complexity of $O(r^3 n)$ operations in case of "dense" rank structures, and $O(r^2 n)$ operations in case the gaps between the structure blocks are of the same order as the corresponding rank indices. In fact, this should not come as a surprise, in the sense that the algorithm in Figure 4.2 has many resemblances to the process of swapping a unitary-weight representation as described in Figure 2.13. Still pursuing this similarity, note that one can use the same techniques as described in subsection 3.1 to obtain a Givens-weight representation of type 2.

Conversely, suppose now that we have given a Givens-weight representation, or more generally a unitary-weight representation, and that we want to find a quasiseparable representation for it. The reader should then revisit Definition 3: denoting with $W_k \in \mathbb{C}^{r_k \times |J_k|}$ the $k$th weight block and with $U_k$ the $k$th unitary transformation of the Givens-weight representation, we recall that the spreading-out process starts by forming

$$U_k^H \begin{bmatrix} W_k \\ 0 \end{bmatrix}.$$

Subsequently, the bottommost $r_{k+1} = |I_{k+1,\mathrm{top}}|$ rows are further spread out by the



(a) Starting situation. We are going to take into account the next transition matrix $T_k$.

(b) Compute a QR-factorization of the matrix formed by $P_k$ and $X_{k+1}T_k$.

(c) The next auxiliary matrix $X_k$ can now be read-off, and we can form the next weight block $X_kQ_{k-1}$.

FIG. 4.2. *Transition from a block quasiseparable to a Givens-weight representation.*

next unitary transformations $U_{k+1}^H, \ldots$. This suggests that we may obtain the quasiseparable parameters by subdividing

$$U_k^H = \left[\begin{array}{c|c} P_k & X \\ T_k & X \end{array}\right],$$

where the blocks $P_k$ and $T_k$ have $|I_k|$ and $|I_{k+1,\mathrm{top}}| = r_{k+1}$ rows, respectively, and $r_k$ columns. The quasiseparable generators $Q_k$ are chosen each time as $Q_k := W_{k+1}$, $k = 0, \ldots, K$.

The dynamics of this algorithm are illustrated in Figure 4.3.

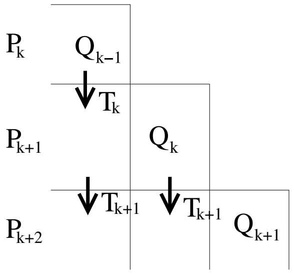


FIG. 4.3. *Schematic picture of a block quasiseparable representation constructed in a "dual" way w.r.t. Figure* 4.1.

Still concerning the algorithm to go from a unitary/Givens-weight to a block quasiseparable representation, note that the above discussion revealed that unitary-weight representations are theoretically equivalent with the matrices $\begin{bmatrix} P_k \\ T_k \end{bmatrix}$ of the quasiseparable representation having orthonormal columns, for each $k$. Thus we see that unitary-weight representations theoretically correspond to the quasiseparable representations in [illegible], following the terminology of [7].

In fact there also exists the notion of [illegible] in [7], meaning that all matrices $\begin{bmatrix} T_k & Q_k \end{bmatrix}$ must have orthonormal rows for each $k$. It can then be argued as before that this notion corresponds to unitary-weight representations that are based on [illegible], rather than row operations.

Thus we see that in a certain sense, the unitary/Givens-weight representation "breaks the symmetry" of block quasiseparable representations by assigning a preference to either row or column operations. This may seem awkward, but as we already observed, this will not prohibit the unitary-weight format from being very [illegible] for the development of algorithms. Moreover, by the fact that the representation is based on [illegible] operations, which are well known to be optimally conditioned w.r.t. the matrix 2-norm, it can be expected that an appropriate use of unitary-weight representations should lead to numerically stable algorithms.

Concerning the differences between block quasiseparable and Givens-weight representations, we should stress that

- The above discussion showed a comparison between block quasiseparable and [illegible] representations. However, the reader should not forget that we are primarily concerned with [illegible] representations, where each unitary transformation $U_k$ has an additional factorization as a sparse product of Givens transformations.[4] Although it may be observed that the Givens-weight representation may deviate by a factor of at most 4 from optimality (since the multiplication of a Givens transformation with a vector of length 2 requires essentially a number of 4 multiplications), it has the advantage that it can [illegible] be used in an efficient way. Notable exceptions where the performance of the Givens-weight representation is also sensible to the distribution of the structure blocks are the [illegible] algorithms described in section 3.

**4.2. *uv*-representations.** In this subsection we briefly consider *uv*-representations for rank structured matrices.[5] Such representations sometimes occur in solution methods for differential and integral equations. Historically, the term "semiseparable matrix" was even introduced in the context of *uv*-representations; see, e.g., [12].

First of all, we must stress that in contrast to Givens-weight and block quasiseparable representations, *uv*-representations do not exist for every rank structured matrix, but instead only for a subclass which we call the class of *uv*-representable matrices. When implementing the QR-algorithm for semiseparable matrices of semiseparability rank 1 using the *uv*-representation for such matrices, for example, this can be considered a severe weakness since the subsequent QR-iterates then converge to a limiting matrix in block upper triangular form, which in general is [illegible] *uv*-representable of rank one anymore; see [17].

Let us now give a formal definition of *uv*-representability.

DEFINITION 10. [illegible] $\mathcal{R}$ [illegible] $r_k =: r$ [illegible] $k$ [illegible] $A \in \mathbb{C}^{m\times n}$ [illegible] *uv*-representable [illegible] $\mathcal{R}$ [illegible]

$$A = uv + A_{\text{completion}}, \tag{4.1}$$

---

[4] This sparsity property constitutes an essential difference between Givens-weight and block quasiseparable representations. However, for completeness, we note that some sparse factorizations in terms of individual Givens transformations were also described in [7, Chapter 14]. But the latter discussion concerns *unitary* rank structured matrices, and we were not able to find there any indication in terms of nonunitary rank structured matrices or algorithmic exploitation.

[5] The matrices allowing such a representation are sometimes called "generator representable matrices" instead of *uv*-representable matrices [17], but we will not use this terminology here since the word "generator" could be confused with, e.g., the quasiseparable generators.

[illegible] $u \in \mathbb{C}^{m\times r}$ [illegible] $v \in \mathbb{C}^{r\times n}$ [illegible] $A_{\text{completion}}$ [illegible] " [illegible] $\mathcal{R}$ [illegible] (4.1) [illegible] $uv$-representation [illegible] $A$

The above definition states that for a matrix to be $uv$-representable, the low rank generators of the different structure blocks of $\mathcal{R}$ should be "compatible" in the sense that they can be completed to a global matrix $uv$ of rank at most $r$.

Now let us give an algorithm to transform a $uv$-representation into a Givens-weight representation. Such an algorithm is trivial and based on the QR-factorization of the matrix $u$ in (4.1). Let us partition $u$ and $v$ as block matrices according to the given distribution of the structure blocks, with block elements $u_k$ and $v_k$, for $k = 0, \dots, K$. Moreover, let us assume that in the process of forming the QR-factorization of $u$, the bottom submatrix $\begin{bmatrix} u_k \\ \vdots \\ u_K \end{bmatrix}$ has just been made upper triangular. Then denoting with $X_k$ the square top block of this upper triangular matrix, the $k$th weight block will be simply $X_k v_{k-1}$. Repeating this process for $k = K, \dots, 1$, at the end we will have obtained a Givens-weight representation for the structured lower triangular part of $A$. Moreover, assuming that the given matrix is square of size $n$, then it is easy to check that (i) the algorithm always leads to an [illegible] Givens-weight representation consisting of not more than $O(rn)$ Givens transformations, and (ii) the algorithm has complexity $O(r^2 n)$, irrespective of the precise distribution of the structure blocks. These properties should be contrasted with the reduction process for quasiseparable representations in subsection 4.1.

## 5. Conclusion.

In this paper we have introduced the notions of unitary-weight and Givens-weight representations for rank structured matrices. It was described, e.g., how the representation can be swapped and how it can be reduced to a lower complexity representation. These results provide a basis for several algorithms using unitary/Givens-weight representations such as QR-factorization, solution of linear systems [4], Hessenberg reduction [6], as well as matrix inversion, explicit and implicit QR-iteration, and so on, to be described in our future work.

## REFERENCES

[1] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen, *Fast stable solver for sequentially semi-separable linear systems of equations*, Lecture Notes Comput. Sci., 2552 (2002), pp. 545–554.

[2] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A.-J. van der Veen, *Fast Stable Solvers for Sequentially Semi-separable Linear Systems of Equations*, Technical report, Department of Mathematics, University of California-Berkeley, Berkeley, CA, 2003.

[3] S. Delvaux and M. Van Barel, *Structures preserved by the QR-algorithm*, J. Comput. Appl. Math., 187 (2006), pp. 29–40.

[4] S. Delvaux and M. Van Barel, *A QR-Based Solver for Rank Structured Matrices*, Technical report TW454, Katholieke Universiteit Leuven, Leuven (Heverlee), Belgium, 2006.

[5] S. Delvaux and M. Van Barel, *Rank structures preserved by the QR-algorithm: The singular case*, J. Comput. Appl. Math., 189 (2006), pp. 157–178.

[6] S. Delvaux and M. Van Barel, *A Hessenberg reduction algorithm for rank structured matrices*, SIAM J. Matrix Anal. Appl., accepted 2007.

[7] P. Dewilde and A.-J. van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.

[8] P. Dewilde and A.-J. van der Veen, *Inner-outer factorization and the inversion of locally finite systems of equations*, Linear Algebra Appl., 313 (2000), pp. 53–100.

[9] Y. Eidelman and I. C. Gohberg, *On a new class of structured matrices*, Integral Equations Operator Theory, 34 (1999), pp. 293–324.

[10] Y. Eidelman and I. C. Gohberg, *A modification of the Dewilde-van der Veen method for inversion of finite structured matrices*, Linear Algebra Appl., 343/344 (2002), pp. 419–450.

[11] Y. Eidelman and I. C. Gohberg, *On generators of quasiseparable finite block matrices*, Calcolo, 42 (2005), pp. 187–214.

[12] I. C. Gohberg, T. Kailath, and I. Koltracht, *Linear complexity algorithms for semiseparable matrices*, Integral Equations Operator Theory, 8 (1985), pp. 780–804.

[13] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[14] E. E. Tyrtyshnikov, *Mosaic ranks for weakly semiseparable matrices*, Large-Scale Scientific Computations of Engineering and Environmental Problems, II, M. Griebel, S. Margenov, and P. Y. Yalamov, eds., Notes Numer. Fluid Mech., 73, Vieweg, Braunschweig, Germany, 2000, pp. 36–41.

[15] M. Van Barel, E. Van Camp, and N. Mastronardi, *Orthogonal similarity transformation into block-semiseparable matrices of semiseparability rank k*, Numer. Linear Algebra Appl., 12 (2005), pp. 981–1000.

[16] R. Vandebril, M. Van Barel, and N. Mastronardi, *An implicit QR algorithm for symmetric semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 625–658.

[17] R. Vandebril, M. Van Barel, and N. Mastronardi, *A note on the representation and definition of semiseparable matrices*, Numer. Linear Algebra Appl., 12 (2005), pp. 839–858.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 4, pp. 1171–1180

# WHEN IS THE ADJOINT OF A MATRIX A LOW DEGREE RATIONAL FUNCTION IN THE MATRIX?[*]

JÖRG LIESEN[†]

**Abstract.** We show that the adjoint $A^+$ of a matrix $A$ with respect to a given inner product is a rational function in $A$, if and only if $A$ is normal with respect to the inner product. We consider such matrices and analyze the McMillan degrees of the rational functions $r$ such that $A^+ = r(A)$. We introduce the McMillan degree of $A$ as the smallest among these degrees, characterize this degree in terms of the number and distribution of the eigenvalues of $A$, and compare the McMillan degree with the normal degree of $A$, which is defined as the smallest degree of a polynomial $p$ for which $A^+ = p(A)$. We show that unless the eigenvalues of $A$ lie on a single circle in the complex plane, the ratio of the normal degree and the McMillan degree of $A$ is bounded by a small constant that depends neither on the number nor on the distribution of the eigenvalues of $A$. Our analysis is motivated by applications in the area of short recurrence Krylov subspace methods.

**Key words.** normal matrices, representation of matrix adjoints, rational interpolation, Krylov subspace methods, short recurrences

**AMS subject classifications.** 15A21, 30C15, 65F10

**DOI.** 10.1137/060675538

## 1. Introduction.

Consider a unitary matrix $A$ with $n \geq 2$ distinct eigenvalues. Since $A$ is normal, its adjoint $A^*$ is a polynomial in $A$ [9, condition 17],

$$A^* \;=\; p(A)\,. \tag{1.1}$$

It has been observed in several publications, e.g., [4, pp. 774–775], that for a unitary matrix $A$, (1.1) does not hold for a polynomial $p$ of "small" degree. In this paper we strengthen this observation by showing that the smallest degree of such polynomial is equal to $n-1$. On the other hand, the [illegible] of a rational function $r = p/q$, where $p$ and $q$ are relatively prime polynomials (i.e., their only common divisor is the constant polynomial 1), is defined as

$$\deg r \;=\; \max\,\{\deg p,\; \deg q\}\,. \tag{1.2}$$

Hence

$$A^* \;=\; r(A)\,, \tag{1.3}$$

where, since $A$ is unitary, $r(z) = 1/z$ is a rational function of McMillan degree one. In summary, the adjoint of a unitary matrix $A$ is a large degree polynomial and a small (McMillan) degree rational function in $A$. The observation that the adjoint of a normal matrix may be represented as a polynomial as well as a rational function in the matrix, and that the degrees of these representations may vastly differ, is more than a curiosity. In fact, it is of great importance for the construction of short recurrence Krylov subspace methods.

---

[*] Received by the editors November 21, 2006; accepted for publication (in revised form) by D. B. Szyld April 11, 2007; published electronically December 7, 2007. This work was supported by the Emmy Noether-Programm of the Deutsche Forschungsgemeinschaft.

http://www.siam.org/journals/simax/29-4/67553.html

[†] Institute of Mathematics, Technical University of Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany (liesen@math.tu-berlin.de).

On the one hand, the fundamental theorem of Faber and Manteuffel [6] shows that if (1.1) holds for a matrix $A$ and a polynomial of degree $s$, then orthogonal Krylov subspace bases for $A$ can be generated by an $(s+2)$-term Arnoldi recurrence (this condition is not only sufficient but also necessary; see [6] or [14] for more details). For a unitary matrix $A$ with $n$ distinct eigenvalues, $A^* = p(A)$ with the smallest possible degree of $p$ being $n-1$. Thus, generating orthogonal Krylov subspace bases for unitary matrices via the Arnoldi process requires a full recurrence.

On the other hand, as shown by Barth and Manteuffel [3, 4], if (1.3) holds for a matrix $A$ and a rational function $r = p/q$, where $p$ and $q$ are relatively prime polynomials of respective degrees $\ell$ and $m$, then orthogonal Krylov subspace bases for $A$ can be generated by a recurrence containing $\ell + m + 2$ terms. The generic type of this recurrence is displayed in [4, equation (4.1)], from which it is easily seen that this recurrence is not of Arnoldi-type (partial necessary conditions for the existence of this recurrence are given in the unpublished report [5]). For stability reasons this (single) recurrence should be implemented in the form of coupled, or multiple recurrences (see [4] for details), but the actual implementation is not important for us here. The important point here is that when $\ell$ and $m$ are small, an orthogonal Krylov subspace basis can be generated by a short recurrence. For a unitary matrix $A$, $r(z) = 1/z$, hence $\ell = 0$ and $m = 1$, so that the length of this recurrence is three. In practical applications one uses coupled two-term recurrences instead of the numerically unstable three-term version. The resulting algorithm, which originally was discovered by Gragg in the context of orthogonal polynomials on the unit circle [8], is called the [illegible]. This algorithm has been used for solving unitary eigenvalue problems (see [17] for a survey) as well as for constructing a minimal residual method for solving linear systems with shifted unitary matrices [11].

For efficiency reasons we would like to use the shortest possible recurrence, and thus we would like to characterize, for a given matrix $A$, the smallest degrees of $p$ and $r$ (if any) such that (1.1) and (1.3), respectively, hold. While the smallest degree of the representation (1.1) has been characterized in the literature (we give here new proofs of some results), comparably little has been done to characterize (1.3). The only related work we are aware of is in the aforementioned papers of Barth and Manteuffel. There, for a given Hermitian positive definite (HPD) matrix $B$, a matrix $A$ is called $B$[illegible] $\ell, m$ , if $A$ is normal with respect to the inner product generated by $B$, and if its adjoint $A^+$ with respect to this inner product satisfies $A^+ = r(A)$, where $r = p/q$ for relatively prime polynomials of respective degrees $\ell$ and $m$, cf. [4, Definition 3.1]. For a given representation $A^+ = r(A)$ with known degrees $\ell$ and $m$, Barth and Manteuffel derive bounds on the maximal number of distinct eigenvalues of $A$ in terms of $\ell$ and $m$, cf. [4, Theorem 3.1], or [3, Theorem 4.1]. However, they provide no characterization of how small or large $\ell$ and $m$ may be for a given matrix $A$, which is the question of interest in this paper (see Remarks 2.4 and 3.7 for further comments on the $B$-normal$(\ell,m)$ matrices).

To allow a rigorous characterization of (1.3), we introduce here the concept of the McMillan degree of $A$, which we define as the smallest McMillan degree of a rational function $r$ such that $A^+ = r(A)$ (section 2). In section 3 we then completely answer the question raised in the title, which, as outlined above, has direct applications in the area of short recurrence Krylov subspace methods. Moreover, we show that unless the eigenvalues of $A$ lie on a single circle in the complex plane, the ratio of the smallest degree of a polynomial representation of $A^+$ (called the normal degree of $A$) and the McMillan degree of $A$ is bounded from above by a small constant (less than five), that depends neither on the number nor on the distribution of the eigenvalues of $A$. In our derivations we apply results from rational interpolation theory, which apparently have not been used in this context before.

## 2. $B$-normal matrices.

Suppose that $A$ is a square matrix and $B$ is an HPD matrix. Throughout the paper we will assume that these matrices are of the same size. The matrix $B$ generates an inner product, $\langle x, y\rangle_B = y^*Bx$, and the adjoint of $A$ with respect to this inner product, or, shortly, the $B$-adjoint of $A$, is $A^+ = B^{-1}A^*B$. If $A^+$ is a polynomial in $A$, then $A$ is said to be $B$[illegible]. This is a straightforward generalization of the common concept of normal matrices, which we here call $I$-normal. Of particular interest is the degree of the polynomial representation of the adjoint.

DEFINITION 2.1. [illegible] $A$ [illegible] $B$ [illegible]

$$A^+ \;=\; p(A)\,, \tag{2.1}$$

[illegible] $p$ [illegible] $s$ [illegible] $A$ [illegible] $s$ [illegible] $B$ [illegible] $B$ [illegible] $s$

The property that $A$ is $B$-normal($s$) is completely characterized in the following result [14, Theorem 3.1].

THEOREM 2.2. [illegible] $A$ [illegible] $B$ [illegible]

1. $A$ [illegible] $B$ [illegible] $s$
2. (a) $A$ [illegible] $A = W\Lambda W^{-1}$ [illegible] $A$ [illegible] $\Lambda$ [illegible]

   (b) [illegible] $W$ [illegible] $A$ [illegible] $B^{-1}$ [illegible] $B^{-1} = WDW^*$ [illegible] $D$ [illegible] $\Lambda$ [illegible]

   (c) [illegible] $p$ [illegible] $s$ [illegible] $p(\Lambda) = \Lambda^*$ [illegible] $s$ [illegible] $p$ [illegible]

In [14] this result is stated only for nonsingular matrices $A$, which is due to the focus of the work in that paper. It is easy to see, however, that the assertion is true also for singular matrices $A$. Using Theorem 2.2, we can characterize all $A$ and $B$ for which $A^+ = r(A)$, where $r$ is a rational function.

LEMMA 2.3. [illegible] $A$ [illegible] $B$ [illegible] $r$ [illegible] $A^+ = r(A)$ [illegible] $r(\lambda) = \overline{\lambda}$ [illegible] $\lambda$ [illegible] $A$ [illegible] $A$ [illegible] $B$ [illegible] $s$ [illegible] $s$ [illegible] $p$ [illegible] $p(\lambda) = \overline{\lambda}$ [illegible] $\lambda$ [illegible] $A$

[illegible]. We adopt the strategy of the proof of [6, Lemma 2]. Let $(\lambda, x)$ be an eigenpair of $A$, $Ax = \lambda x$. Then $A^+x = r(A)x = r(\lambda)x$, so that

$$r(\lambda)\,\langle x,x\rangle_B \;=\; \langle r(\lambda)x, x\rangle_B \;=\; \langle A^+x, x\rangle_B \;=\; \langle x, Ax\rangle_B \;=\; \langle x, \lambda x\rangle_B \;=\; \overline{\lambda}\,\langle x,x\rangle_B\,,$$

from which we receive $r(\lambda) = \overline{\lambda}$.

Now suppose that there is a nontrivial Jordan block associated with $\lambda$. Then there exists a nonzero vector $y$ such that $(A-\lambda I)y = x$. But then

$$\begin{aligned} \langle Ay, x\rangle_B = \langle \lambda y + x, x\rangle_B \;&=\; \lambda\, \langle y, x\rangle_B + \langle x, x\rangle_B\,, \\ \langle Ay, x\rangle_B = \langle y, A^+ x\rangle_B \;&=\; \langle y, \overline{\lambda} x\rangle_B \;=\; \lambda\, \langle y, x\rangle_B\,, \end{aligned}$$

which means that $\langle x, x\rangle_B = 0$. This contradiction shows that $A$ is diagonalizable, i.e., that (2a) of Theorem 2.2 holds.

If $(\eta, y)$ is another eigenpair of $A$ with $\eta \neq \lambda$, then

$$\lambda\, \langle x, y\rangle_B \;=\; \langle \lambda x, y\rangle_B \;=\; \langle Ax, y\rangle_B \;=\; \langle x, A^+ y\rangle_B \;=\; \langle x, \overline{\eta} y\rangle_B \;=\; \eta\, \langle x, y\rangle_B\,.$$

Since $\lambda \neq \eta$ we must have $\langle x, y\rangle_B = 0$, which shows that the eigenvectors of $A$ form a complete $B$-orthogonal set. In particular, when we consider the diagonalizable matrix $A$ as in (2a) of Theorem 2.2, then $W^*BW = D$, where $D$ is HPD and block diagonal, showing that $B$ is as stated in (2b) of Theorem 2.2.

Finally, the polynomial $p$ in (2c) of Theorem 2.2 is the uniquely determined interpolation polynomial of smallest degree that satisfies $p(\lambda) = \overline{\lambda}$ for all eigenvalues $\lambda$ of $A$. ☐

[illegible] 2.4. According to Lemma 2.3, the existence of a representation of the form $A^+ = r(A)$, where $r$ is a rational function, implies that $A$ is $B$-normal($s$). Therefore the assumption that $A$ be $B$-normal in the definition of the $B$-normal($\ell$,$m$) matrices of Barth and Manteuffel, cf. [3, Definition 4.2] or [4, Definition 3.1], is redundant.

The converse of Lemma 2.3 is obviously true as well: If $A$ is $B$-normal($s$), then there exists a rational function, namely $r = p$ from (2c) in Theorem 2.2, such that $A^+ = r(A)$. We therefore have the following corollary.

COROLLARY 2.5. [illegible] $A$ [illegible] $B$ [illegible] $A^+$ [illegible] $A$ [illegible] $A$ [illegible]

[illegible] $A$ [illegible] $A = W\Lambda W^{-1}$ [illegible] $B$ [illegible] $A^+$ [illegible] $A$ [illegible] (2b) [illegible] 2.2 [illegible] $B$ [illegible] $A^+ = r(A)$ [illegible] $r$ [illegible] $r(\Lambda) = \Lambda^*$

[illegible] Only the necessity part in the last sentence remains to be shown. Let $B$ be any matrix as characterized in (2b) of Theorem 2.2, i.e., $B = W^{-*}DW^{-1}$. Then

$$A^+ = B^{-1}A^*B = (WD^{-1}W^*)(W^{-*}\Lambda^*W^*)(W^{-*}DW^{-1}) = W\Lambda^*W^{-1} = r(A)\,,$$

where in the last equation we have used that $r(\Lambda) = \Lambda^*$. ☐

By Corollary 2.5, for a nondiagonalizable matrix $A$ there exists no HPD matrix $B$ such that the corresponding $A^+$ is a rational function in $A$. We therefore can restrict our attention to diagonalizable matrices. The last part of the corollary shows that if, for some HPD matrix $B$, $A^+$ is a rational function in $A$, $A^+ = r(A)$, then $r$ is completely determined by the eigenvalues of $A$. We use the following concepts in our further development.

DEFINITION 2.6. [illegible] $A$ [illegible]

1. [illegible] $p$ [illegible] $p(\lambda) = \overline{\lambda}$ [illegible] $\lambda$ [illegible] $A$ [illegible] $A$ [illegible] $d_p(A)$
2. [illegible] $r$ [illegible] $r(\lambda) = \overline{\lambda}$ [illegible] $\lambda$ [illegible] $A$ [illegible] $A$ [illegible] $d_r(A)$

We immediately observe that $d_r(A) \leq d_p(A) \leq n-1$, where $n$ is the number of distinct eigenvalues of $A$.

Let us put the degrees $d_p(A)$ and $d_r(A)$ into the picture of short recurrence Krylov subspace methods that is described in the introduction: On the one hand, if $A$ is normal with respect to an HPD matrix $B$ and $d_p(A) = s$, then $B$-orthogonal Krylov subspace bases for $A$ can be generated with an $(s+2)$-term Arnoldi recurrence [6]. On the other hand, if for an HPD matrix $B$ the $B$-adjoint of $A$ satisfies $A^+ = r(A)$, where $r = p/q$ for relatively prime polynomials $p$ and $q$ of respective degrees $\ell$ and $m$, so that $d_r(A) = \deg r = \max\{\ell, m\}$, then $B$-orthogonal Krylov subspace bases for $A$ can be generated using a nonstandard recurrence containing $\ell + m + 2 \leq 2d_r(A) + 2$ terms [4]. If $d_r(A) \ll d_p(A)$, then the nonstandard recurrence is significantly more efficient than the standard Arnoldi recurrence. It is therefore of great practical interest to characterize the (diagonalizable) matrices $A$ for which $d_r(A) \ll d_p(A)$.

**3. Characterization of the McMillan degree of $A$.** We will study the McMillan degree of a diagonalizable matrix $A$ using results from rational interpolation theory. The results we employ were originally developed by Antoulas and Anderson [2] and are summarized in Antoulas' book [1, Chapter 4.5].

Let $\lambda_1, \ldots, \lambda_n$ be the [illegible] eigenvalues of $A$. We want to determine a rational function $r = p/q$, where $p$ and $q$ are relatively prime polynomials, such that $r(\lambda_j) = \overline{\lambda}_j$, $j = 1, \ldots, n$. We assume $n \geq 2$, as otherwise the problem is trivial. If there exists such a rational function of McMillan degree $m$, then $m$ is called an [illegible]. By definition, the smallest admissible McMillan degree is equal to $d_r(A)$.

Consider the array $\mathbb{P}$ containing the interpolation points $(\lambda_j, \overline{\lambda}_j)$, $j = 1, \ldots, n$,

$$\mathbb{P} \;=\; \{(\lambda_j, \overline{\lambda}_j) \,:\, j = 1, \ldots, n\}\,. \tag{3.1}$$

We choose an integer $n_1$, $1 \leq n_1 < n$, and partition $\mathbb{P}$ into two disjoint subarrays $\mathbb{J}$ and $\mathbb{I}$,

$$\mathbb{J} \;=\; \{(\lambda_j, \overline{\lambda}_j) \,:\, j = 1, \ldots, n_1\}\,, \qquad \mathbb{I} \;=\; \{(\lambda_j, \overline{\lambda}_j) \,:\, j = n_1 + 1, \ldots, n\}\,.$$

For notational convenience, we now write $\mu_j \equiv \lambda_{j+n_1}$ for $j = 1, \ldots, n - n_1$. Then the [illegible] $L$ corresponding to the arrays $\mathbb{J}$ and $\mathbb{I}$ is defined by

$$L \;=\; [l_{i,j}]_{i=1,\ldots,n-n_1,\, j=1,\ldots,n_1}, \quad \text{where} \quad l_{i,j} \;=\; \frac{\overline{\mu}_i - \overline{\lambda}_j}{\mu_i - \lambda_j}\,. \tag{3.2}$$

Note that $L$ is of size $(n - n_1) \times n_1$. Moreover, the [illegible] $\mathbb{P}$ is defined as

$$\operatorname{rank} \mathbb{P} \;=\; \max_{L}\, \{\, \operatorname{rank} L \,\}\,, \tag{3.3}$$

where the maximum is taken over all possible Löwner matrices, which can be formed from $\mathbb{P}$ by partitioning into two subarrays as described above, cf. [1, Definition 4.51].

A similar construction can be made for any subarray of interpolation points. More precisely, we may take any $\mathbb{Q} \subset \mathbb{P}$ containing at least two points, partition $\mathbb{Q}$ into two disjoint subarrays, and form the corresponding Löwner matrix according to (3.2). In this way we can form Löwner matrices from $\mathbb{P}$ that are of size $k_1 \times k_2$ with $k_1 + k_2 < n$.

Theorem 3.1 (cf. [1, Theorem 4.55 and Corollary 4.56]). [illegible] $\mathbb{P}$ [illegible] (3.1) [illegible] $m$

(1) [illegible] $2m < n$ [illegible] $m \times m$ [illegible] $\mathbb{P}$ [illegible] $r = p/q$ [illegible] $p$ [illegible] $q$ [illegible] $m$ [illegible] $r(\lambda_j) = \overline{\lambda}_j$ $j = 1, \ldots, n$ [illegible] $m$ [illegible] $n - m$

(2) [illegible] $n - m$ [illegible] $r = p/q$ [illegible] $p$ [illegible] $q$ [illegible] $n - m$ [illegible] $r(\lambda_j) = \overline{\lambda}_j$ $j = 1, \ldots, n$

The following is a straightforward consequence.

Corollary 3.2. [illegible] $A$ [illegible] $n$ [illegible] $d_r(A) \leq \lceil n/2 \rceil$ [illegible] $n \in \{2, 3\}$ [illegible] $d_r(A) = 1$

Having characterized the cases $n = 2$ and $n = 3$, we will now focus on matrices with at least four distinct eigenvalues.

Lemma 3.3. [illegible] $\lambda_1, \ldots, \lambda_4$ [illegible] $\mathbb{P}$ [illegible] (3.1) [illegible] $\operatorname{rank} \mathbb{P} = 1$ [illegible] $\lambda_1, \ldots, \lambda_4$ [illegible]

[illegible]. We partition $\mathbb{P}$ into two subarrays $\mathbb{J}$ and $\mathbb{I}$ containing two interpolation points each. The corresponding Löwner matrix is

$$L \;=\; \left[ \frac{\overline{\mu}_i - \overline{\lambda}_j}{\mu_i - \lambda_j} \right]_{i,j=1,2},$$

giving

$$\det L \;=\; \frac{\overline{\mu}_1 - \overline{\lambda}_1}{\mu_1 - \lambda_1} \, \frac{\overline{\mu}_2 - \overline{\lambda}_2}{\mu_2 - \lambda_2} \;-\; \frac{\overline{\mu}_2 - \overline{\lambda}_1}{\mu_2 - \lambda_1} \, \frac{\overline{\mu}_1 - \overline{\lambda}_2}{\mu_1 - \lambda_2} \,.$$

Hence $\det L = 0$, if and only if

$$\frac{(\mu_1 - \lambda_1)\,(\mu_2 - \lambda_2)}{(\mu_1 - \lambda_2)\,(\mu_2 - \lambda_1)} \;\in\; \mathbb{R}\,. \tag{3.4}$$

We denote by $\widehat{\mathbb{C}}$ the extended complex plane. Recall that a circle in $\widehat{\mathbb{C}}$ is either a true circle in the complex plane or a line in the complex plane with the point at infinity adjoined. In (3.4) we replace $\mu_2$ by a variable $z$, and consider the function

$$f(z) \;=\; \frac{(\mu_1 - \lambda_1)\,(z - \lambda_2)}{(\mu_1 - \lambda_2)\,(z - \lambda_1)}\,.$$

The function $f(z)$ is the unique Moebius transformation satisfying

$$f(\lambda_1) = \infty\,, \quad f(\lambda_2) = 0\,, \quad f(\mu_1) = 1\,.$$

Now realize that through the points $\lambda_1, \lambda_2, \mu_1$ passes one and only one circle $\mathcal{C}$ in $\widehat{\mathbb{C}}$. Since the Moebius transformation $f$ conformally maps circles in $\widehat{\mathbb{C}}$ onto circles in $\widehat{\mathbb{C}}$, we see that $f(\mathcal{C}) = \mathbb{R} \cup \{\infty\}$, and, in particular, $f(\mu_2) \in \mathbb{R}$, if and only if $\mu_2 \in \mathcal{C}$ (see, e.g., [15, Chapter 3] for more on Moebius transformations). Consequently, $L$ is singular if and only if $\lambda_1, \lambda_2, \mu_1, \mu_2$ lie on the same circle in $\widehat{\mathbb{C}}$, i.e., if and only if these points in the complex plane are either collinear or concyclic, which completes the proof. $\square$

Using this lemma we can characterize the diagonalizable matrices of McMillan degree one. To do so, we recall that a matrix has rank $k$ if and only if it has a nonsingular $k \times k$ submatrix and all its $(k+1) \times (k+1)$ submatrices are singular (see, e.g., [10, pp. 12–13]).

LEMMA 3.4. [illegible] $A$ [illegible] $d_r(A) = 1$ [illegible] $A$ [illegible]

[illegible]. Let $L$ be any Löwner matrix with at least two rows and columns formed from the array $\mathbb{P}$ corresponding to the $n \geq 4$ distinct eigenvalues of $A$. Clearly, $\operatorname{rank} L \geq 1$.

If the eigenvalues of $A$ are either collinear or concyclic, then Lemma 3.3 shows that every $2 \times 2$ submatrix of $L$ is singular. Therefore, $\operatorname{rank} L = 1$, which shows that $\operatorname{rank} \mathbb{P} = 1$. Since $2 < n$, and every $1 \times 1$ Löwner matrix that can be formed from $\mathbb{P}$ is nonsingular, case (1) in Theorem 3.1 applies, showing that $d_r(A) = 1$.

On the other hand, if the eigenvalues of $A$ are neither collinear nor concyclic, then by Lemma 3.3 there exists a $2 \times 2$ submatrix of $L$ that is nonsingular. Hence $\operatorname{rank} L \geq 2$, which implies that $\operatorname{rank} \mathbb{P} \geq 2$, and hence $d_r(A) \geq 2$. ☐

If $A$ is any diagonalizable matrix with $n \geq 2$ distinct eigenvalues that all are collinear, then Corollary 3.2 and Lemma 3.4 show that $d_r(A) = 1$. These matrices are also known to be $B$-normal(1) for some HPD matrix $B$ (cf. [14, Theorem 3.3] and the references given there), and thus they satisfy $d_p(A) = 1$.

More interesting is the class of the diagonalizable matrices with $n \geq 3$ distinct eigenvalues that all are concyclic. For such matrices $A$, Corollary 3.2 and Lemma 3.4 show that $d_r(A) = 1$. Moreover, case (1) in Theorem 3.1 shows that the rational function $r$ of McMillan degree one that satisfies $r(\lambda) = \overline{\lambda}$ for all eigenvalues $\lambda$ is uniquely determined. In fact, this function can be easily computed. Suppose that the eigenvalues of $A$ are given by

$$\lambda_j \;=\; \rho e^{i\varphi_j} + \zeta\,, \quad j = 1, \ldots, n\,,$$

where $\rho \in \mathbb{R} \setminus \{0\}$ and $\zeta \in \mathbb{C}$ do not depend on $j$, while $\varphi_j \in [0, 2\pi)$. Then

$$r(z) \;=\; \frac{\overline{\zeta} z + (\rho^2 - |\zeta|^2)}{z - \zeta}$$

satisfies $r(\lambda_j) = \overline{\lambda}_j$, $j = 1, \ldots, n$. Clearly, $r$ is not a polynomial. Case (1) in Theorem 3.1 also shows that the next smallest admissible McMillan degree is $n - 1$. Apparently, a corresponding rational function is the uniquely determined (Lagrange) interpolation polynomial $p$ that satisfies $p(\lambda) = \overline{\lambda}$ for all eigenvalues $\lambda$. This means that $d_p(A) = n - 1$.

It is easy to see that, for a diagonalizable matrix $A$, $d_r(A)$ is equal to the smallest possible McMillan degree of a rational function $r$ such that the eigenvalues of $A$ are zeros of the function $r(z) - \overline{z}$. When $n \geq 4$ and the eigenvalues of $A$ are neither collinear nor concyclic, Lemma 3.4 implies that $d_r(A) \geq 2$. Hence in this case we search for a rational function $r$ of (smallest possible) $\deg r \geq 2$, such that the eigenvalues of $A$ are zeros of $r(z) - \overline{z}$. The following result summarizes what is known about the zeros of such functions.

THEOREM 3.5.

(1) [illegible] $p(z) - \overline{z}$ [illegible] $p$ [illegible] $s \geq 2$ [illegible] $3s - 2$ [illegible] $s \geq 2$ [illegible] $p$ [illegible] $s$ [illegible] $p(z) - \overline{z}$ [illegible] $3s - 2$ [illegible]

(2) [illegible] $r(z)-\overline{z}$ [illegible] $r$ [illegible] $s \geq 2$ [illegible] $5s-5$ [illegible] $s \geq 2$ [illegible] $r$ [illegible] $s$ [illegible] $r(z)-\overline{z}$ [illegible] $5s-5$ [illegible]

The bounds in (1) and (2) have been shown in [13] and [12], respectively. The corresponding sharpness results have been shown in [7] and [16]. Using these bounds we can prove the following result.

Theorem 3.6. [illegible] $A$ [illegible] $n \geq 4$ [illegible]

(1) [illegible] $d_r(A) = d_p(A) = 1$

(2) [illegible] $d_r(A) = 1$ [illegible] $d_p(A) = n-1$

(3) [illegible] $d_r(A) \geq \lfloor n/5 + 1 \rfloor$ $d_p(A) \geq \lfloor (n+2)/3 \rfloor$ [illegible]

$$1 \,\leq\, \frac{d_p(A)}{d_r(A)} \,\leq\, 5\,\frac{n-1}{n+5} \,<\, 5\,. \tag{3.5}$$

[illegible]. Cases (1) and (2) were shown above, so only case (3) needs to be proven. Here the eigenvalues are neither collinear nor concyclic, and thus by Lemma 3.4 we must have $d_r(A) \geq 2$. From case (2) in Theorem 3.5 we know that any function of the form $r(z) - \overline{z}$, where $r$ is a rational function of $\deg r \geq 2$, may have at most $5 \deg r - 5$ zeros. Since any function for which the McMillan degree of $A$ is attained must have (at least) $n$ distinct zeros, we must have $n \leq 5 d_r(A) - 5$, and thus $d_r(A) \geq \lfloor n/5 + 1 \rfloor$. The lower bound on $d_p(A)$ follows in a similar way from case (1) in Theorem 3.5. Finally, the leftmost and rightmost inequalities in (3.5) are straightforward, while the middle inequality follows from the lower bound on $d_r(A)$ and from noting that $d_p(A) \leq n-1$. □

Using Theorem 2.2, Corollary 3.2, and Theorem 3.6 we can derive the following well-known result: There exists an HPD matrix $B$ with respect to which a matrix $A$, with at least two distinct eigenvalues is normal of degree one, if and only if $A$ is diagonalizable and has collinear eigenvalues (cf. [14, Theorem 3.3] and the references given there). Here we have given a new proof of this result using rational interpolation theory and conformal mappings.

A surprising fact shown by Theorem 3.6 is that the ratio $d_p(A)/d_r(A)$ is bounded from above by five, unless the eigenvalues of $A$ are concyclic, in which case the ratio is equal to $n-1$. In this sense, the diagonalizable matrices with concyclic eigenvalues form a very special class.

Theorem 3.6 also shows that if the eigenvalues of a diagonalizable matrix $A$ are neither collinear nor concyclic, then $d_r(A)$ is small, if and only if $A$ has only a small number (at most $5d_r(A) - 5$) of distinct eigenvalues.

[illegible] 3.7. A related observation is made after the statement of [4, Theorem 3.1], but it is not fully justified from the theory presented there. According to Barth and Manteuffel, their result "says that if $A$ is $B$-normal($\ell$,$m$) and either $\ell$ or $m$ greater than 1, then $A$ has a relatively small number of distinct eigenvalues" [4, p. 775]. However, in terms of [4, Definition 3.1], any unitary matrix $A$ with $n \geq 3$ distinct eigenvalues is $I$-normal(0,1) [illegible] $I$-normal($n-1$,0). Hence, for $\ell = n-1 > 1$ and $m = 0$, $A$ is $B$-normal($\ell$,$m$), but $A$ may have arbitrarily many distinct eigenvalues. The confusion is caused by the lack of uniqueness of the "smallest degrees" $\ell$ and $m$. In general, there exist no "simultaneously smallest" $\ell$ and $m$ for which $A^+ = r(A)$ with $r = p/q$ for relatively prime polynomials of respective degrees $\ell$ and $m$.

We next show by examples that the two weak inequalities in (3.5) cannot be improved in general. First, consider the lower bound on $d_p(A)/d_r(A)$. This bound is attained if and only if a rational function $r$ of smallest possible McMillan degree, which satisfies $r(\lambda) = \overline{\lambda}$ for all eigenvalues $\lambda$ of $A$, is a polynomial (this always holds when the eigenvalues of $A$ are collinear, cf. case (1) in Theorem 3.6, where $d_r(A) = d_p(A) = 1$). Consider a diagonalizable matrix $A$ with $n = 4$ distinct eigenvalues given by

$$\lambda_1 = 1 + \sqrt{1/2}, \quad \lambda_2 = 1 - \sqrt{1/2}, \quad \lambda_3 = i\sqrt{1/2}, \quad \lambda_4 = -i\sqrt{1/2}.$$

The polynomial

$$p(z) \; = \; z^2 - z + \frac{1}{2}$$

is the unique polynomial of smallest possible degree that satisfies $p(\lambda_j) = \overline{\lambda}_j$, $j = 1, \ldots, 4$, so that $d_p(A) = 2$. On the other hand, $d_r(A) \leq 2$ by Corollary 3.2, and since the four eigenvalues are neither collinear nor concyclic, Lemma 3.4 implies that $d_r(A) = 2$, showing that the lower bound in (3.5) is attained.

To give an example that the upper bound is attained, consider any diagonalizable matrix $A$ with $n = 5$ distinct eigenvalues that are neither collinear nor concyclic. By Corollary 3.2, $d_r(A) \leq \lceil 5/2 \rceil = 2$, and by Lemma 3.4, $d_r(A) > 1$, showing that $d_r(A) = 2$. Suppose that the five eigenvalues are

$$\lambda_1 = 0, \quad \lambda_2 = 1, \quad \lambda_3 = 2, \quad \lambda_4 = i, \quad \lambda_5 = -i.$$

Obviously, these are neither collinear nor concyclic. An elementary computation (that may be performed by any computer algebra package) shows that the unique polynomial $p$ of smallest possible degree that satisfies $p(\lambda_j) = \overline{\lambda}_j$, $j = 1, \ldots, 5$, is given by

$$p(z) \; = \; \frac{3}{5}z - \frac{3}{5}z^2 + \frac{8}{5}z^3 - \frac{3}{5}z^4,$$

so that $d_p(A) = 4$. Therefore, $d_p(A)/d_r(A) = 2$, showing that the weak upper bound in (3.5) is attained.

Finally, we remark that it may be possible to extend our approach to give an alternative proof of the sharpness of the bound of [12] on the maximal number of zeros of $r(z) - \overline{z}$, where $r$ is rational with $\deg r \geq 2$ (cf. case (2) in Theorem 3.5). For example, let five distinct complex numbers be given, such that any four of them are neither collinear nor concyclic. Then, by case (1) in Theorem 3.1, there exists a unique rational function $r$ of $\deg r = 2$, so that the five complex numbers are zeros of $r(z) - \overline{z}$. This function $r$ can be explicitly computed along the lines of [1, pp. 105–107], and it attains the bound of [12].

**Acknowledgments.** Thanks to Jurjen Duintjer Tebbens, Daniel Szyld, and Petr Tichý for comments that helped to improve the presentation of this paper.

## REFERENCES

[1] A. C. Antoulas, *Approximation of large-scale dynamical systems*, with a foreword by Jan C. Willems, Advances in Design and Control 6, SIAM, Philadelphia, 2005.

[2] A. C. Antoulas and B. D. O. Anderson, *On the scalar rational interpolation problem*, IMA J. Math. Control Info., 3 (1986), pp. 61–88.

[3] T. L. Barth and T. A. Manteuffel, *Conjugate gradient algorithms using multiple recursions*, in Proceedings of the AMS-IMS-SIAM Summer Research Conference, Seattle, 1995, L. Adams and J. L. Nazareth, eds., SIAM, Philadelphia, 1996, pp. 107–123.

[4] T. Barth and T. Manteuffel, *Multiple recursion conjugate gradient algorithms.* I. *Sufficient conditions*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 768–796.

[5] T. Barth and T. Manteuffel, *Multiple Recursion Conjugate Gradient Algorithms.* II. *Necessary Conditions*, unpublished manuscript, 2000.

[6] V. Faber and T. Manteuffel, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.

[7] L. Geyer, *Sharp Bounds for the Valence of Certain Harmonic Polynomials*, in Proceedings of the American Mathematical Society, accepted; also available online at arXiv:math.CV/0510539, 2005.

[8] W. B. Gragg, *Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and Gaussian quadrature on the unit circle*, J. Comput. Appl. Math., 46 (1993), pp. 183–198.

[9] R. Grone, C. R. Johnson, E. M. de Sá, and H. Wolkowicz, *Normal matrices*, Linear Algebra Appl., 87 (1987), pp. 213–225.

[10] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[11] C. F. Jagels and L. Reichel, *A fast minimal residual algorithm for shifted unitary matrices*, Numer. Linear Algebra Appl., 1 (1994), pp. 555–570.

[12] D. Khavinson and G. Neumann, *On the number of zeros of certain rational harmonic functions*, Proc. Amer. Math. Soc., 134 (2006), pp. 1077–1085.

[13] D. Khavinson and G. Świątek, *On the number of zeros of certain harmonic polynomials*, Proc. Amer. Math. Soc., 131 (2003), pp. 409–414.

[14] J. Liesen and Z. Strakoš, *On Optimal Short Recurrences for Generating Orthogonal Krylov Subspace Bases*, SIAM Rev., accepted.

[15] T. Needham, *Visual Complex Analysis*, The Clarendon Press, Oxford University Press, New York, 1997.

[16] S. H. Rhie, *$n$-point Gravitational Lenses with* $5(n-1)$ *Images*, Technical report, available online at arXiv:astro-ph/0305166, 2003.

[17] D. S. Watkins, *Some perspectives on the eigenvalue problem*, SIAM Rev., 35 (1993), pp. 430–471.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 4, pp. 1181–1190

# ON THE INDEX OF CONDITIONAL STABILITY OF STABLE INVARIANT LAGRANGIAN SUBSPACES*

ANDRÉ C. M. RAN† AND LEIBA RODMAN‡

**Abstract.** Given a nondegenerate sesquilinear inner product on a finite dimensional complex vector space, or a nondegenerate symmetric or skewsymmetric inner product on finite dimensional real vector space, subspaces that are simultaneously Lagrangian and invariant for a selfadjoint or a skewadjoint matrix with respect to the inner product are considered. The rate of conditional stability of such subspaces is studied, under small perturbations of both the inner product and the matrix. The concept of conditional stability (in contrast with unconditional stability) presupposes that one considers only those perturbed matrix and inner product for which the existence of invariant Lagrangian subspaces can be guaranteed a priori. Open problems regarding the index (= exact rate) of conditional stability are stated. Several inaccurate statements in the authors' previous works concerning the index are made precise. Finally, an application is given to conditional stability of hermitian solutions of continuous type algebraic Riccati equations.

**Key words.** invariant subspaces, Lagrangian subspaces, indefinite inner product, stability

**AMS subject classifications.** Primary, 15A63; Secondary, 47A15, 47A55

**DOI.** 10.1137/060678531

## 1. Introduction.

Let $\mathbb{F}$ denote either the field of complex numbers $\mathbb{C}$ or the field of real numbers $\mathbb{R}$ and let $H \in \mathbb{F}^{m\times m}$ be a hermitian (symmetric in the real case) invertible matrix. Then a matrix $A \in \mathbb{F}^{m\times m}$ is called $H$-*selfadjoint* if $HA$ is hermitian, i.e., $HA = A^*H$, or $HA = A^TH$ in the real case, where the superscript $^T$, resp., $^*$, denotes the transposed, resp., conjugate transposed, matrix or vector. We consider also $H$-skewadjoint matrices $A$, i.e., such that $HA$ is skewadjoint. (Note that only in the real case the $H$-skewadjoint matrices form an essentially different class, because in the complex case $A$ is $H$-skewadjoint if and only if $iA$ is $H$-selfadjoint.) In the real case it is of interest to study the classes of matrices $A \in \mathbb{R}^{m\times m}$ with the properties that $HA = \pm A^TH$, where $H \in \mathbb{R}^{m\times m}$ is a given invertible skewsymmetric matrix ($m$ must be even for such $H$ to exist); if $HA = A^TH$, the matrix $A$ is said to be $H$-*selfadjoint*, and if $HA = -A^TH$, the matrix $A$ is said to be $H$-*skewadjoint*. (In the literature the terms $H$-Hamiltonian and $H$-skew-Hamiltonian are used as well.) The theory and applications of these classes of matrices is a well studied area of linear algebra, see, for example, books [13], [6], [7], [10], and recent expository papers [11], [12].

From now on we assume that $m = 2n$ is even, and let $H \in \mathbb{F}^{m\times m}$ be an invertible hermitian matrix (in the complex case) or invertible symmetric or skewsymmetric matrix (in the real case). We say that a subspace $\mathcal{M} \subseteq \mathbb{F}^{2n}$ is $H$-*Lagrangian* if $\dim \mathcal{M} = n$ and

$$y^*Hx = 0 \quad (y^THx = 0 \text{ in the real case}) \quad \text{for all} \quad x, y \in \mathcal{M}.$$

---

*Received by the editors December 25, 2006; accepted for publication (in revised form) by P. Benner April 24, 2007; published electronically December 7, 2007.
http://www.siam.org/journals/simax/29-4/67853.html

†Afdeling Wiskunde, Faculteit der Exacte Wetenschappen, Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands (ran@cs.vu.nl).

‡Department of Mathematics, College of William and Mary, P.O. Box 8795, Williamsburg, VA 23187-8795 (lxrodm@math.wm.edu). The research of this author was partially supported by NSF grant DMS-0456625 and by Summer Grant from the College of William and Mary.

If $A$ is $H$-selfadjoint or $H$-skewadjoint, then $A$-invariant $H$-Lagrangian subspaces are of particular interest in many applications. Note that such subspaces do not exist for every $H$-selfadjoint or $H$-skewadjoint matrix need. Denote by $\mathcal{IL}(A, H)$ the (possibly empty) set of all $A$-invariant $H$-Lagrangian subspaces.

We now give a key definition of conditional $\alpha$-stability, a concept that was introduced and studied in various guises in [25], [22], [24], [21], [23]. A well-known notion of the [illegible] $(\mathcal{M}, \mathcal{N})$ between two subspaces $\mathcal{M}, \mathcal{N} \subseteq \mathbb{F}^{2n}$ will be used:

$$\operatorname{gap}(\mathcal{M}, \mathcal{N}) := \|P_{\mathcal{M}} - P_{\mathcal{N}}\|,$$

where $P_{\mathcal{M}}$, resp., $P_{\mathcal{N}}$, is the orthogonal projection on $\mathcal{M}$, resp., $\mathcal{N}$, and the operator norm (= the largest singular value) $\|\cdot\|$ is used throughout. See, for example, [8] for more information on the gap function.

Fix $\alpha \geq 1$, and let $A \in \mathbb{F}^{2n \times 2n}$ be an $H$-selfadjoint or $H$-skewadjoint matrix. A subspace $\mathcal{M} \in \mathcal{IL}(A, H)$ is called [illegible] $\alpha$ [illegible] if there exist $\delta > 0$ and $K > 0$ with the following properties: If $H' \in \mathbb{F}^{2n \times 2n}$ is hermitian and $A'$ is $H'$-selfadjoint or $H'$-skewadjoint as the case may be, and

$$\|A - A'\| + \|H - H'\| < \delta,$$

and if $\mathcal{IL}(A', H') \neq \emptyset$, then there exists $\mathcal{M}' \in \mathcal{IL}(A', H')$ such that

$$\operatorname{gap}(\mathcal{M}, \mathcal{M}') \leq K \left(\|A - A'\| + \|H - H'\|\right)^{1/\alpha}.$$

If in the above definition $H$ is kept fixed, i.e., the additional restriction $H' = H$ is imposed, then the definition of a conditionally $H$-$\alpha$-stable subspace $\mathcal{M} \in \mathcal{IL}(A, H)$ is obtained. One can show (we omit details) that the concept of conditional $H$-$\alpha$-stability, although formally weaker than that of conditional $\alpha$-stability, is in fact equivalent to it (compare [16], [17]). Note also that if $\mathcal{M}$ is conditionally $\alpha$-stable, then $\mathcal{M}$ is conditionally $\beta$-stable for every $\beta > \alpha$.

The concept of conditional $\alpha$-stability is one of many related notions of stability that have been studied in the literature starting with [4], [1], [2]. Although on the face of it the conditional $\alpha$-stability seems to be a rather contrived notion, it does play a role in important applications, for example, classical $H_\infty$ control. In this application, a solution to the control problem involves certain $A(\gamma)$-invariant $H$-Lagrangian subspaces, where $A(\gamma)$ is a real $H$-skewadjoint matrix parameterized by a positive parameter $\gamma$, and where $H \in \mathbb{R}^{2n \times 2n}$ is a fixed invertible skewsymmetric matrix. In some situations, the optimal solution corresponds to the minimal value $\gamma_0$ of $\gamma$ for which $A(\gamma)$-invariant $H$-Lagrangian subspaces exist (this approach to $H_\infty$ control was developed in [5] in the context of solutions of certain algebraic Riccati equations; see also [10, Chapter 20] for more details). Clearly, conditional stability is an appropriate tool for perturbation analysis in the vicinity of the optimal solution, since $\mathcal{IL}(A(\gamma), H) = \emptyset$ for $\gamma < \gamma_0$.

In the present paper we continue the investigation of conditional $\alpha$-stability initiated in [22], see also [23]. As it turns out, there are inaccuracies in several statements in [20], [22], [23] (all of them can be traced to the same source). We correct the statements, add a few additional results and applications, and formulate open problems. This theme will be further developed in [14] in the context of symplectic matrices.

**2. Conditional $\alpha$-stability: Complex case.** We state and prove here the main result on conditional $\alpha$-stability of invariant Lagrangian subspaces for $H$-selfadjoint matrices in the complex case. As before, $H \in \mathbb{C}^{2n\times 2n}$ is a fixed invertible hermitian matrix. Denote by $\mathcal{R}_\lambda(A) = \mathrm{Ker}\,(A-\lambda I)^{2n}$ the root subspace of a matrix $A \in \mathbb{C}^{2n\times 2n}$ corresponding to its eigenvalue $\lambda$.

Theorem 2.1. [illegible] $A \in \mathbb{C}^{2n\times 2n}$ [illegible] $H$ [illegible] $A$ [illegible]

$$\alpha_+ = \max\{m_1, \dots, m_r\},$$

[illegible] $m_1, \dots, m_r$ [illegible] $A$ [illegible] $\alpha_+ = 1$ [illegible] $A$ [illegible]

$$\alpha_- = \max\{2, m_1 - 1, \dots, m_r - 1\}$$

[illegible] $\alpha_- = 1$ [illegible] $A$ [illegible]

(I) [illegible] $\alpha_+$ [illegible] $A$ [illegible] $H$ [illegible] $\mathcal{M}$ [illegible] $\mathcal{M}$ [illegible]

$$\{0\} \neq \mathcal{M} \cap \mathcal{R}_\lambda(A) \neq \mathcal{R}_\lambda(A)$$

[illegible] $\lambda$ [illegible] $A$ [illegible] $\lambda$ [illegible] $1$ [illegible]

(1) [illegible] $\lambda$ [illegible] $\alpha_+$.

(2) [illegible] $\lambda$ [illegible] $\alpha_+ + 1$ [illegible]

$$k := \dim\,(\mathcal{M} \cap \mathcal{R}_\lambda(A)),$$

[illegible] $k$ [illegible] $(\alpha_+ + 1)$ [illegible]

(II) [illegible] $\beta$ [illegible] $A$ [illegible] $H$ [illegible] $\beta$ [illegible] $1 \leq \beta < \alpha_-$

Note that we always have $\alpha_- \leq \alpha_+$. Note also that the hypothesis that the geometric multiplicities of $A$ corresponding to real eigenvalues are all equal to 1 is not necessary for existence of conditionally stable $A$-invariant $H$-Lagrangian subspaces; see [17] for more information.

A particular case of Theorem 2.1 deserves special attention, namely the $A$-invariant $H$-Lagrangian subspace $\mathcal{M}_0$ with the property that the spectrum of the restriction of $A$ to $\mathcal{M}_0$ is in the closed upper half plane. Note that this subspace is unique under the hypotheses of Theorem 2.1.

Corollary 2.2. [illegible] 2.1 [illegible] $\mathcal{M}_0$ [illegible] $\alpha_+$ [illegible] $\beta$ [illegible] $1 \leq \beta < \alpha_-$

An analogous corollary holds for the $A$-invariant $H$-Lagrangian subspace with the spectrum of the restriction of $A$ to the subspace being in the closed lower half plane.

It has been claimed in [20], [22, Theorem 6.5], [23, Theorem 3.14] (in the context of $\alpha$-stability of solutions of algebraic Riccati equations) that there exist conditionally $\alpha_-$-stable $A$-invariant $H$-Lagrangian subspaces, under the hypotheses of Theorem 2.1. However, the proof given in [22], [23] falls short of proving this claim (but see Theorem 2.3). Therefore, we can state an open problem.

Problem 1. [illegible] $\alpha_-$ [illegible] $A$ [illegible] $H$ [illegible] 2.1

[illegible] 2.1. The proof of part (II) follows from [22, Lemma 6.8], using the criterion for $\beta$-stability of invariant subspaces of general complex matrices (see [24] for details).

For part (I), we first use the localization principle (see [17], [16]). Note that in [16] the localization principle is stated for the stability property of invariant Lagrangian subspaces; the same considerations yield the localization principle also in our context of conditional $\alpha$-stability. Thus, we need only to consider two cases: (a) $A$ has no real eigenvalues; (b) $A$ has only one real eigenvalue, perhaps of large multiplicity. In the case (a), we have $\alpha_+ = \alpha_- = 1$, and the spectral $A$-invariant subspace corresponding to the eigenvalues in the open upper half place (or open lower half plane) is $H$-Lagrangian and conditionally 1-stable. In the case (b), note that there is a unique $A$-invariant $H$-Lagrangian subspace $\mathcal{M}$, and its dimension is $n$. The result of [25] implies that there exist $\delta > 0$ and $K > 0$ such that every complex matrix $B$ with $\|B - A\| < \delta$ has the property that every $n$-dimensional $B$-invariant subspace $\mathcal{N}$ satisfies the inequality

$$\operatorname{gap}(\mathcal{M}, \mathcal{N}) \leq \|B - A\|^{1/\alpha_+}.$$

In particular, if $B$ is also $H'$-selfadjoint and $\mathcal{IL}(B, H') \neq \emptyset$, then one such $n$-dimensional $B$-invariant subspace will be $H'$-Lagrangian, and conditional $\alpha_+$-stability of $\mathcal{M}$ follows. $\square$

Let $A$ be as in Theorem 2.1, and let $\mathcal{M}$ be an $A$-invariant $H$-Lagrangian subspace. We say that $\alpha_0 \geq 1$ is the [illegible] of $\mathcal{M}$ if $\mathcal{M}$ is conditionally $\alpha_0$-stable but is not conditionally $\alpha$-stable for any $\alpha$ with $1 \leq \alpha < \alpha_0$. The index is obviously unique, but its existence is not obvious even when $\mathcal{M}$ is conditionally $\alpha$-stable for some $\alpha \geq 1$. Indeed, a priori it is not clear that such $\mathcal{M}$ is conditionally $\alpha_0$-stable, where $\alpha_0$ is the greatest lower bound of the (nonempty) set of all $\alpha$'s with the property that $\mathcal{M}$ is conditionally $\alpha$-stable (see Problem 2 below).

Note that generally speaking Theorem 2.1 does not provide the index of conditional stability. One particular case when the index can be obtained using the theorem is presented in the next result. It will be convenient to introduce the following notation: For a positive integer $m$, and any integer $k$, $0 \leq k \leq m$, define

$$\alpha_{\mathbb{C}}(m,k) := \begin{cases} 1 & \text{if } k = 0 \text{ or } k = m, \\ m-1 & \text{if } 1 \leq k \leq m-1 \text{ and } \exists\ k \text{ distinct } m\text{th roots of unity} \\ & \text{whose sum is 0,} \\ m & \text{in all other cases.} \end{cases}$$

THEOREM 2.3. [illegible] $A \in \mathbb{C}^{2n\times 2n}$ [illegible] $H$ [illegible] $A$ [illegible] $\alpha_0 = 1$ [illegible] $A$ [illegible] $\alpha_0 = 2$ [illegible]

[illegible] $\mathcal{M}$ [illegible] $A$ [illegible] $H$ [illegible] $\mathcal{M}$ [illegible] $\alpha$ [illegible] $\alpha$ [illegible]

[illegible]

$$\{0\} \neq \mathcal{M} \cap \mathcal{R}_\lambda(A) \neq \mathcal{R}_\lambda(A)$$

[illegible] $\lambda$ [illegible] $A$ [illegible] $\lambda$ [illegible] 1 [illegible] $\mathcal{M}$ [illegible] $\alpha_0$ [illegible]

$$\max_{j=1,2,\ldots,r} \left\{\alpha_{\mathbb{C}}\left(\dim \mathcal{R}_{\lambda}(A), \dim\left(\mathcal{R}_{\lambda}(A) \cap \mathcal{M}\right)\right)\right\}, \tag{2.1}$$

[illegible] $\lambda_1, \ldots, \lambda_r$ [illegible] $A$ [illegible] (2.1) [illegible] 1 [illegible] (2.1) $\lambda_1, \ldots, \lambda_r$ [illegible] $A$ [illegible]

[illegible] Note that $\alpha_0 = \alpha_+ = \alpha_-$, where $\alpha_\pm$ are taken from Theorem 2.1. By using the localization principle again, two cases need to be considered: (a) $A$ has only one real eigenvalue, perhaps of large multiplicity; (b) $A$ has no real eigenvalues. In case (b), we complete the proof by using a result proved in [17] (see also [23, Theorem 3.5]). The result says that (assuming the $H$-selfadjoint matrix $A$ has no real eigenvalues) an $A$-invariant $H$-Lagrangian subspace $\mathcal{M}$ is conditionally stable if and only if the intersection of $\mathcal{M}$ with the spectral invariant subspace $\mathcal{R}_+(A)$ of $A$ corresponding to the eigenvalues in the open upper half plane is stable as an $A$-invariant subspace. Moreover, the index of conditional stability of $\mathcal{M}$ coincides with the index of stability of $\mathcal{M} \cap \mathcal{R}_+(A)$, and the latter was computed in [24] leading to formula (2.1). In case (a), just use Theorem 2.1. □

It follows from Theorem 2.3 that the index of conditional stability of the $A$-invariant $H$-Lagrangian subspace with spectrum in the closed upper half (resp. lower half) plane is $\alpha_0$ under the hypotheses and the notation of Theorem 2.3.

In view of Theorem 2.3, the following open problem (a part of which is in fact a particular case of a very general and seemingly intractable problem [23, Problem 3.6]) is suggested.

PROBLEM 2. [illegible] $A$ [illegible] 2.1 [illegible] $\mathcal{M}$ [illegible] $A$ [illegible] $H$ [illegible]

$$\{0\} \neq \mathcal{M} \cap \mathcal{R}_{\lambda}(A) \neq \mathcal{R}_{\lambda}(A)$$

[illegible] $\lambda$ [illegible] $A$ [illegible] $\lambda$ [illegible] 1

(a) [illegible] $\mathcal{M}$

(b) [illegible]

**3. Conditional $\alpha$-stability: Real case.** In this section, $H$ is an invertible real symmetric or skewsymmetric $2n \times 2n$ matrix. Invariant Lagrangian subspaces of real $H$-selfadjoint and $H$-skewadjoint matrices and their various stability properties had been studied in [18], [19]. However, not much is known about conditional $\alpha$-stability, except the facts that can be derived from results on other types of stability in [18], [19]. Thus we have the following problem.

PROBLEM 3. [illegible] $A$ [illegible] $H$ [illegible] $H$ [illegible] $A$ [illegible] $H$ [illegible] $\alpha \geq 1$ [illegible] $\alpha$ [illegible] $A$ [illegible] $H$ [illegible]

In this formulation, the problem is probably intractable. We present certain analogues of Theorems 2.1 and 2.3 for $H$-skewadjoint matrices, under the basic hypothesis that pure imaginary and zero eigenvalues of $A$ have geometric multiplicity one, and state a (hopefully more tractable) particular case of Problem 3.

We need some additional notation. A finite set of complex numbers $\{\zeta_1, \ldots, \zeta_m\}$ will be called [illegible] if $\zeta_1 + \cdots + \zeta_m = 0$, and the nonreal elements of the set can be arranged in pairs of complex conjugate numbers. For two integers $k$

and $m$, with $0 \le k \le m$, $m > 0$, we define $\alpha_{\mathbb{R}}(m,\ k)$ as follows: $\alpha_{\mathbb{R}}(m,\ k) = m$ in the following three cases: (i) $0 < k < m$, $m$ is odd and there is no zero sum selfconjugate set of $k$ distinct $m$th roots of 1, (ii) $m$ is even and $k$ is odd, (iii) $m$ is even and divisible by 4, $k$ is also even but not divisible by 4, and there is no zero sum selfconjugate set of $k$ distinct $m$th roots of $-1$. We define $\alpha_{\mathbb{R}}(m,\ k) = 1$ if $k = 0$ or $k = m$. In all other cases we define $\alpha_{\mathbb{R}}(m,\ k) = m - 1$.

Denote by $\mathcal{R}_{a\pm ib}(A)$ the root subspace corresponding to a pair of nonreal complex conjugate eigenvalues $a \pm ib$, $a \in \mathbb{R}$, $b \in \mathbb{R} \setminus \{0\}$, of a real $m \times m$ matrix $A$:

$$\mathcal{R}_{a\pm ib}(A) := \operatorname{Ker}(A^2 - 2aA + (a^2 + b^2)I)^m \subseteq \mathbb{R}^m.$$

THEOREM 3.1. [illegible] $A \in \mathbb{R}^{2n\times 2n}$ [illegible] $H$ [illegible] $H \in \mathbb{R}^{2n\times 2n}$ [illegible] $A$ [illegible] $A$ [illegible]

$$\alpha_+ = \max\{m_1, \ldots, m_r\},$$

[illegible] $m_1, \ldots, m_r$ [illegible] $A$ [illegible] $\alpha_+ = 1$ [illegible] $A$ [illegible]

$$\alpha_- = \max\{2, m_1 - 1, \ldots, m_r - 1\}$$

[illegible] $\alpha_- = 1$ [illegible] $A$ [illegible]

(I) [illegible] $\alpha_+$ [illegible] $A$ [illegible] $H$ [illegible] $\mathcal{M}$ [illegible] $\mathcal{M}$ [illegible] (X) [illegible] (Y)

(X) [illegible]

$$\{0\} \ne \mathcal{M} \cap \mathcal{R}_\lambda(A) \ne \mathcal{R}_\lambda(A)$$

[illegible] $\lambda$ [illegible] $A$ [illegible] $\lambda$ [illegible] 1 [illegible]

(1) [illegible] $m$ [illegible] $\lambda$ [illegible] $\alpha_+$.

(2) [illegible] $m$ [illegible] $\lambda$ [illegible] $\alpha_+ + 1$ [illegible]

$$k := \dim\left(\mathcal{M} \cap \mathcal{R}_\lambda(A)\right),$$

[illegible] $\alpha_+ = \alpha_{\mathbb{R}}(m,\ k)$.

[illegible] $m$ [illegible] $k$

(Y) [illegible] $a \pm ib$ $(a, b \in \mathbb{R} \setminus \{0\})$ [illegible] $A$ [illegible]

$$\{0\} \ne \mathcal{R}_{a\pm ib}(A) \cap \mathcal{M} \ne \mathcal{R}_{a\pm ib}(A),$$

[illegible] $a+ib$ [illegible] $a-ib$ [illegible] $m$ [illegible]

(3) $m \le \alpha_+$.

(4) $m = \alpha_+ + 1$ [illegible] $\alpha_+ = \alpha_{\mathbb{C}}\left(m, \frac{\dim \mathcal{R}_{a\pm ib}(A)\cap\mathcal{M}}{2}\right)$.

(II) [illegible] $\beta$ [illegible] $A$ [illegible] $H$ [illegible] $\beta$ [illegible] $1 \le \beta < \alpha_-$

The proof is parallel to that of Theorem 2.1. We need a criterion for $\beta$-stability of real invariant subspaces of real matrices (without symmetries). Such a criterion was established in [21].

Analogously to Corollary 2.2 we can derive the conditional stability result for the $A$-invariant $H$-Lagrangian subspace $\mathcal{M}_\ell$ with the property that the spectrum of the restriction of $A$ to $\mathcal{M}_\ell$ is in the closed left half plane. Note that this subspace is unique under the hypotheses of Theorem 3.1.

Corollary 3.2. [illegible] 3.1 [illegible] $\mathcal{M}_\ell$ [illegible] $\alpha_+$ [illegible] $\beta$ [illegible] $1 \leq \beta < \alpha_-$

Under the additional hypothesis that $m_1 = \cdots = m_r = 2$, an analogue of Theorem 2.3 may be obtained. We leave the formulation of this analogue to the interested reader.

We state an open problem analogous to Problem 2.

Problem 4. [illegible] $A$ [illegible] 3.1 [illegible] $\mathcal{M} \subseteq \mathbb{R}^{2n}$ [illegible] $A$ [illegible] $H$ [illegible]

$$\{0\} \neq \mathcal{M} \cap \mathcal{R}_\lambda(A) \neq \mathcal{R}_\lambda(A)$$

[illegible] $\lambda$ [illegible] $A$ [illegible] $\lambda$ [illegible] 1 [illegible]

$$\{0\} \neq \mathcal{R}_{a\pm ib}(A) \cap \mathcal{M} \neq \mathcal{R}_{a\pm ib}(A)$$

[illegible] $a \pm ib$ [illegible] $A$ [illegible] $a \pm ib$ [illegible] 1

[illegible] $\mathcal{M}$ [illegible]

**4. Applications: Algebraic Riccati equations.** Theorems 2.1 and 3.1 have many important well-known applications, several of them studied in [22], [23].

We give a detailed statement for the algebraic Riccati equations application. The literature on this topic is voluminous; we mention only books [3], [10], [15], [9], where more information, applications, and references are found.

We start with the complex case. Consider the algebraic Riccati equation

$$XDX - XA - A^*X - C = 0, \tag{4.1}$$

where $A$, $D = D^*$, $C = C^*$ are given $n \times n$ complex matrices, and $X = X^*$ is to be found. The Hamiltonian matrix

$$M := i \begin{bmatrix} A & -D \\ -C & -A^* \end{bmatrix} \tag{4.2}$$

of the Riccati equation plays a key role in the theory.

For a given $\alpha \geq 1$, a (hermitian) solution $X$ of (4.1) is called [illegible] $\alpha$ [illegible] if there exist $\varepsilon > 0,\ K > 0$ such that every equation with coefficients in $\mathbb{C}^{n\times n}$

$$X\widetilde{D}X - X\widetilde{A} - \widetilde{A}^*X - \widetilde{C} = 0, \tag{4.3}$$

with $\widetilde{D} = \widetilde{D}^*,\ \widetilde{C} = \widetilde{C}^*$, and

$$\|D - \widetilde{D}\| + \|A - \widetilde{A}\| + \|C - \widetilde{C}\| < \varepsilon$$

has a (hermitian) solution $Y \in \mathbb{C}^{n\times n}$ such that

$$\|X-Y\| \le K(\|D-\widetilde{D}\| + \|A-\widetilde{A}\| + \|C-\widetilde{C}\|)^{\frac{1}{\alpha}},$$

[illegible] (4.3) [illegible]

THEOREM 4.1. [illegible] $D$ [illegible] $(A,D)$ [illegible] $\lambda$ [illegible] $A$ [illegible] $\mathcal{R}_\lambda(A)$ [illegible] $\mathcal{R}_{-\overline{\lambda}}(A)$ [illegible]

$$\text{Range}\,[D, AD, \dots, A^{n-1}D]$$

[illegible] $(A,D)$ [illegible] $\lambda_1,\dots,\lambda_r$ [illegible] $M$ [illegible] $m_1,\dots,m_r$ [illegible]

$$\alpha_+ = \max\{m_1,\dots,m_r\},$$

[illegible] $\alpha_+ = 1$ [illegible] $M$ [illegible]

$$\alpha_- = \max\{2, m_1-1,\dots,m_r-1\}$$

[illegible] $\alpha_- = 1$ [illegible] $M$ [illegible]

(I) [illegible] $\alpha_+$ [illegible] $X$ [illegible] (4.1) [illegible] 2.1 [illegible] $A$ [illegible] $M$ [illegible]

$$\mathcal{M} := \text{Range}\begin{bmatrix} I \\ X \end{bmatrix}.$$

(II) [illegible] $\beta$ [illegible] (4.1) [illegible] $\beta$ [illegible] $1 \le \beta < \alpha_-$

(III) [illegible] $X_0$ [illegible] $X_0'$ [illegible] $M$ [illegible] $\left[\begin{smallmatrix} I \\ X_0 \end{smallmatrix}\right]$ [illegible] $\left[\begin{smallmatrix} I \\ X_0' \end{smallmatrix}\right]$ [illegible] $\alpha_+$ [illegible]

The result of Theorem 4.1 follows immediately from Theorem 2.1 and Corollary 2.2, using the well-known fact (see [10], for example) that under the hypotheses of Theorem 4.1, the (hermitian) solutions $X$ of are in one-to-one correspondence with $M$-invariant $i\left[\begin{smallmatrix} 0 & I \\ -I & 0 \end{smallmatrix}\right]$-Lagrangian subspaces.

PROBLEM 5. [illegible] 4.1 [illegible] $\alpha_-$ [illegible] $X$ [illegible] (4.1)

We introduce the index of conditional stability analogously to the index for invariant Lagrangian subspaces. Namely, we say that $\alpha_0 \ge 1$ is the [illegible] of a solution $X$ if $X$ is conditionally $\alpha_0$-stable but is not conditionally $\alpha$-stable for any $\alpha$ with $1 \le \alpha < \alpha_0$. Again, we have the analogue of Theorem 2.3.

THEOREM 4.2. [illegible] 4.1 [illegible] $M$ [illegible] $\alpha_0 = 1$ [illegible] $M$ [illegible] $\alpha_0 = 2$ [illegible]

[illegible] $X$ [illegible] (4.1) [illegible] $X$ [illegible] $\alpha$ [illegible] $\alpha \ge 1$ [illegible]

[illegible]

$$\{0\} \neq \left(\text{Range}\begin{bmatrix} I \\ X \end{bmatrix}\right) \cap \mathcal{R}_\lambda(M) \neq \mathcal{R}_\lambda(M)$$

[illegible] $\lambda$ [illegible] $M$ [illegible] $\lambda$ [illegible] 1 [illegible] $\mathcal{M}$ [illegible] $\alpha_0$ [illegible]

$$\max_{j=1,2,\ldots,r}\left\{\alpha_{\mathbb{C}}\left(\dim\mathcal{R}_\lambda(M),\dim\left(\mathcal{R}_\lambda(M)\cap\left(\operatorname{Range}\begin{bmatrix} I \\ X\end{bmatrix}\right)\right)\right)\right\}, \tag{4.4}$$

[illegible] $\lambda_1,\ldots,\lambda_r$ [illegible] $M$ [illegible] (4.4) [illegible] 1

We remark that analogously to Theorems 4.1 and 4.2, the corresponding results for the real case can be obtained from Theorem 3.1 and the real analogue of Theorem 2.3 (under the additional hypothesis that $m_1 = \cdots = m_r = 2$). In the real case, one uses the matrices $\left[\begin{smallmatrix} A & D \\ C & -A^T\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} 0 & I \\ -I & 0\end{smallmatrix}\right]$ instead of $M$ (given by (4.2)) and $i\left[\begin{smallmatrix} 0 & I \\ -I & 0\end{smallmatrix}\right]$, respectively. We omit the details.

Finally, we indicate corrections that should be made in [22, Theorems 6.5 and 7.4], [23, Theorem 3.14]. Namely, in part (i) of these theorems $\alpha_0$ should be replaced with $\max\{m_1,\ldots,m_r\}$ if $M$ has real eigenvalues (or if $T$ has unimodular eigenvalues in [22, Theorem 7.4]). Also, in [22, Theorem 6.9] in the statement "there exist conditionally $\alpha_0$-stable real hermitian solutions" the number $\alpha_0$ should be replaced with $\max\{m_1,\ldots,m_r\}$ if $M_r$ has pure imaginary or zero eigenvalues; an analogous correction should be made in [20]. In the minimal factorization result of [23, Theorem 3.31], in the statement "there exists a conditionally $\alpha_0$-stable symmetric factorization of $W$" the number $\alpha_0$ should be replaced with $\max\{m_1,\ldots,m_r,n_1,\ldots,n_s\}$ provided $W$ has real poles or zeros (or both).

**Acknowledgment.** We thank C. Mehl and V. Mehrmann for useful discussions concerning the subject matter of this paper. L. Rodman gratefully acknowledges hospitality and support during his visit at the Vrije Universiteit in Amsterdam.

## REFERENCES

[1] H. Bart, I. Gohberg, and M. Kaashoek, *Stable factorizations of monic matrix polynomials and stable invariant subspaces*, Integral Equations Operator Theory, 1 (1978), pp. 496–517.

[2] H. Bart, I. Gohberg, and M. Kaashoek, *Minimal factorization of matrix and operator functions*, Operator Theory: Advances and Applications, Vol. 1, Birkhäuser Verlag, Basel, 1979.

[3] S. Bittanti, A. J. Laub, and J. C. Willems, eds., *The Riccati Equation*, Communications and Control Engineering Series, Springer-Verlag, Berlin, 1991.

[4] S. Campbell and J. Daughtry, *The stable solutions of quadratic matrix equations*, Proc. Amer. Math. Soc., 74 (1979), pp. 19–23.

[5] J. C. Doyle, K. Glover, P. P. Khargonekar, and F. Francis, *State-space solutions to standard $H_2$ and $H_\infty$ control problems*, IEEE Trans. Automat. Control, 34 (1989), pp. 831–847.

[6] I. Gohberg, P. Lancaster, and L. Rodman, *Matrices and Indefinite Scalar Products*, Operator Theory: Advances Appl. Vol. 8, Birkhäuser Verlag, Basel, 1983.

[7] I. Gohberg, P. Lancaster, and L. Rodman, *Indefinite Linear Algebra and Applications*, Birkhäuser, Basel, 2005.

[8] I. Gohberg, P. Lancaster, and L. Rodman, *Invariant Subspaces of Matrices with Applications*, John Wiley and Sons, New York, 1986; republished in Classics Appl. Math., SIAM, Philadelphia, 2006.

[9] V. Ionescu. C. Oară, and M. Weiss, *Generalized Riccati Theory and Robust Control. A Popov Function Approach*, John Wiley and Sons, Chichester, UK, 1999.

[10] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon, Oxford, UK, 1995.

[11] P. Lancaster and L. Rodman, *Canonical forms for Hermitian matrix pairs under strict equivalence and congruence*, SIAM Rev., 47 (2005), pp. 407–443.

[12] P. Lancaster and L. Rodman, *Canonical forms for symmetric/skew-symmetric real matrix pairs under strict equivalence and congruence*, Linear Algebra Appl., 406 (2005), pp. 1–76.

[13] A. I. Mal'cev, *Foundations of Linear Algebra*, W. H. Freeman, San Francisco, 1963 (translation from Russian).

[14] C. Mehl, V. Mehrmann, A. C. M. Ran, and L. Rodman, *Perturbation Analysis of Lagrangian Invariant Subspaces of Symplectic Matrices*, to appear in Linear and Multilinear Algebra.

[15] V. L. Mehrmann, *The Autonomous Linear Quadratic Control Problem. Theory and Numerical Solution*, Lecture Notes in Control and Inform. Sci. 163, Springer-Verlag, Berlin, 1991.

[16] A. C. M. Ran and L. Rodman, *Stability of neutral invariant subspaces and stable symmetric factorizations*, Integral Equations Operator Theory, 6 (1983), pp. 536–571.

[17] A. C. M. Ran and L. Rodman, *Stability of invariant maximal semidefinite subspaces*, I, Linear Algebra Appl., 62 (1984), pp. 51–86.

[18] A. C. M. Ran and L. Rodman, *Stability of invariant Lagrangian subspaces*, I, in Topics in Operator Theory, I. Gohberg, ed., Oper. Theory Adv. Appl. 32, Birkhäuser, Basel, 1988, pp. 181–218.

[19] A. C. M. Ran and L. Rodman, *Stability of invariant Lagrangian subspaces*, II, in The Gohberg Anniversary Collection Vol. I, Oper. Theory Adv. Appl. 40, Birkhäuser, Basel, 1989, pp. 391–425.

[20] A. C. M. Ran and L. Rodman, *Rate of stability of hermitian solutions of algebraic Riccati equations*, in Proceedings of the 5th SIAM Conference on Applied Linear Algebra, Snowbird, UT, J. G. Lewis, ed., SIAM, Philadelphia, 1994, pp. 3–6.

[21] A. C. M. Ran and L. Rodman, *The rate of convergence of real invariant subspaces*, Linear Algebra Appl., 207 (1994), pp. 197–224.

[22] A. C. M. Ran and L. Rodman, *Rate of stability of solutions of matrix polynomial and quadratic equations*, Integral Equations Operator Theory, 27 (1997), pp. 71–102.

[23] A. C. M. Ran and L. Rodman, *A class of robustness problems in matrix analysis*, in Interpolation Theory, Systems Theory and Related Topics, Oper. Theory Adv. Appl. 134, Birkhauser, Basel, 2002, pp. 337–383.

[24] A. C. M. Ran, L. Rodman, and A. L. Rubin, *Stability index of invariant subspaces of matrices*, Linear and Multilinear Algebra, 36 (1993), pp. 27–39.

[25] A. C. M. Ran and L. Roozemond, *On strong $\alpha$-stability of invariant subspaces of matrices*, in The Gohberg Anniversary Collection Vol. I, Oper. Theory Adv. Appl. 40, Birkhäuser, Basel, 1989, pp. 427–435.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1191–1217

# RECURSIVE SOLUTION OF CERTAIN STRUCTURED LINEAR SYSTEMS*

ANDRÉ KLEIN† AND PETER SPREIJ‡

**Abstract.** We provide explicit representations of the null space $\mathcal{S}$ of adjoints of companion-related matrices and of certain rectangular generalized Vandermonde matrices of block Toeplitz type which are encountered in the Fisher information matrix of time series processes. A formula for the right-inverse of this class of matrices $A$ is provided which allows one to express the solution of the system $Ax = b$ as $x = A^{-}b + \mathcal{S}$. The formulas can be easily turned into solution algorithms.

**Key words.** linear systems, coefficient matrix, null space, generalized Vandermonde matrix, Toeplitz matrix

**AMS subject classification.** 15A06

**DOI.** 10.1137/060656115

## 1. Introduction.

The subject of this paper is concerned with a recursive solution of new linear systems of equations. The following two linear systems of equations are investigated:

$$\mathcal{K}_\nu(\sigma)X = \mathcal{E} \tag{1.1}$$

and

$$\mathcal{M}_\tau(\rho)Y = \mathcal{R}. \tag{1.2}$$

The coefficient matrices in (1.1) and (1.2) have the form

$$\mathcal{K}_\nu(\sigma) = \left(\frac{d^\nu}{dz^\nu}\left(u_q(z)u_q^{*\top}(z)\right), \frac{d^{\nu-1}}{dz^{\nu-1}}\left(u_q(z)u_q^{*\top}(z)\right), \ldots, u_q(z)u_q^{*\top}(z)\right)_{z=\sigma}$$

and

$$\mathcal{M}_\tau(\rho) = \left(\frac{d^\tau}{dz^\tau}\left(\operatorname{adj}(zI - C_p)\right), \frac{d^{\tau-1}}{dz^{\tau-1}}\left(\operatorname{adj}(zI - C_p)\right), \ldots, \operatorname{adj}(zI - C_p)\right)_{z=\rho},$$

where $\mathcal{K}_\nu(\sigma) \in \mathbb{R}^{q\times q(\nu+1)}$ and $\mathcal{M}_\tau(\rho) \in \mathbb{R}^{p\times p(\tau+1)}$. The companion matrix $C_p \in \mathbb{R}^{p\times p}$ is given by

$$C_p = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & 1 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & 0 & 1 \\ -c_p & & & -c_2 & -c_1 \end{pmatrix}, \tag{1.3}$$

---

*Received by the editors April 3, 2006; accepted for publication (in revised form) by D. A. Bini May 3, 2007; published electronically December 7, 2007.
http://www.siam.org/journals/simax/29-4/65611.html

†Department of Quantitative Economics, University of Amsterdam, Roetersstraat 11, 1018 WB Amsterdam, The Netherlands (aklein@fee.uva.nl).

‡Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands (spreij@science.uva.nl).

where $\rho$ is an eigenvalue of $C_p$ with algebraic multiplicity $\tau + 1$, $\top$ denotes the transpose, $\mathrm{adj}(X)$ denotes the adjoint of matrix $X$, $\sigma$, $q$, and $\nu$ are arbitrary scalar values. Further, we have

$$u_q(z) = (1, z, \ldots, z^{q-1})^\top \quad \text{and} \quad u_q^*(z) = (z^{q-1}, \ldots, 1)^\top. \tag{1.4}$$

Here $X$ and $Y$ are matrices of size $q\,(\nu + 1) \times \ell$ and $p\,(\tau + 1) \times h$, respectively, while $\mathcal{E}$ and $\mathcal{R}$ have size $q \times \ell$ and $p \times h$, respectively. The coefficient matrices in (1.1) are rectangular generalized Vandermonde matrices of block Toeplitz type and in (1.2) they are adjoints of companion-related matrices. The linear equations studied in this paper are extracted from [5], where the Fisher information matrix of a stationary time series process is interconnected with a solution to a Stein equation. The matrix $\mathcal{E}$ is the Fisher information matrix of a stationary time series process, whereas matrix $\mathcal{R}$ is a solution to a Stein equation for an extended version of $\mathcal{M}_\tau(\rho)$. The matrices $X$ and $Y$ are equal and this enables the interconnections to be successfully implemented. In this paper, stationary processes do not play any role, contrary to [5]. However, it is worth noticing that the interconnection between Toeplitz forms and stationary processes has been extensively studied in [3].

In [5], $q$ is the degree of a polynomial $d_q(z)$ in $z \in \mathbb{C}$, $\sigma$ is a root of polynomial $d_q(z)$ with algebraic multiplicity $\nu + 1$. In other words, $q$, $\sigma$, and $\nu$ are interconnected through polynomial $d_q(z)$, whereas in this paper $q$, $\sigma$, and $\nu$ are arbitrary scalar values with no link to a common polynomial and $q$, $\nu > 0$. The algorithm derived in [5] constructs a vector belonging to the null space of $\mathcal{K}_\nu(\sigma)$, which requires matrix multiplications.

A property proved in [6] is used in [5] to derive an algorithm for the kernel of $\mathcal{M}_\tau(\rho)$, it concerns an interconnection between $\mathrm{adj}(zI - C_p)$ and the basis vector $u_p(z)$, this holds for $p = q$, $\sigma = \rho$, $\nu = \tau$ and when $\rho$ is an eigenvalue of $C_p$. The vectors $y \in \mathrm{Ker}\ (\mathcal{M}_\tau(\rho))$ and $x \in \mathrm{Ker}\ (\mathcal{K}_\tau(\rho))$, where $\mathrm{Ker}(X)$ is the kernel of the matrix $X$, are then interconnected. Consequently, the algorithm of the null space of $\mathcal{M}_\tau(\rho)$ given by vector $y$ is based on the algorithm of the null space of $\mathcal{K}_\tau(\rho)$ expressed by vector $x$. The computation of the vector $y$ involves an inversion of a lower triangular and Toeplitz matrix. However, this is combined with $p\tau$ matrix multiplications of the inverted matrix with the corresponding vector $x \in \mathrm{Ker}\ (\mathcal{K}_\tau(\rho))$. This is in agreement with the dimension of the null space of $\mathcal{M}_\tau(\rho)$.

In this paper the approach is different, (1.1) and (1.2) are two different linear systems of equations without a common matrix, and we develop a new algorithm for the null space of the coefficient matrices $\mathcal{K}_\nu(\sigma)$ and $\mathcal{M}_\tau(\rho)$ independently.

A solution of the linear systems of (1.1) and (1.2) is considered when $q = \nu + 1$ and $p = \tau + 1$. In this case, the newly developed algorithms for the null spaces and right-inverses are equivalent for both coefficient matrices. The appropriate right-inverse is expressed in terms of a generalized Vandermonde matrix. A new algorithm is also developed for the kernel of $\mathcal{K}_\nu(\sigma)$ for the case $q > \nu + 1$. The newly displayed algorithms for the null space do not require matrix multiplications and matrix inversions. The main computational exercise consists of evaluating factorials and binomial coefficients, the latter can be computed by applying the Pascal triangle, combined with recursions that consist of addition of two vectors. However, the problem set forth in this paper is algebraical. The purpose is to write a solution of new linear systems of equations as a function of $z$ and the problem studied is therefore not numerical. For that purpose one will subsequently consider the coefficient matrix $\mathcal{K}_\nu(z)$. When we consider the coefficient matrix $\mathcal{M}_\tau(z)$, for technical reasons that shall be specified in section 4, we will then consider the case $z = \rho$.

When $q = \nu + 1$ and $p = \tau + 1$, the representation of the null space $\mathrm{Ker}(\mathcal{M}_\tau(\rho))$ is obtained by simply transposing certain matrices in the representation of the null space Ker $(\mathcal{K}_\nu(\sigma))$. This means that when the algorithm of Ker $(\mathcal{M}_\tau(\rho))$ needs to be evaluated one can use the algorithm for the null space Ker $(\mathcal{K}_\nu(\sigma))$. Contrary to the corresponding algorithm displayed in [5], where a matrix inversion and matrix multiplications are involved, there is no need for a computational exercise of any kind when the algorithm set forth in this paper is applied.

Another fundamental difference with the approach in [5] is that the algorithms developed in this paper cover the entire span of the null spaces of $\mathcal{K}_\nu(\sigma)$ and $\mathcal{M}_\tau(\rho)$ and not just a vector as in [5].

Consequently, we may apply these results to provide explicit expressions of the solutions to the systems (1.1) and (1.2); more specifically, for $q = \nu + 1$ and $p = \tau + 1$ we have

$$X = (\mathcal{K}_\nu(\sigma))^{-} \mathcal{E} + \mathcal{W}(\sigma) \text{ with } \mathcal{W}(\sigma) \in \text{ Ker } (\mathcal{K}_\nu(\sigma)), \tag{1.5}$$

$$Y = (\mathcal{M}_\tau(\rho))^{-} \mathcal{R} + \mathcal{L}(\rho) \text{ with } \mathcal{L}(\rho) \in \text{ Ker } (\mathcal{M}_\tau(\rho)). \tag{1.6}$$

The similarity of the null spaces of the coefficient matrices in (1.1) and (1.2) is interesting. It implies a connection between adjoints of companion-related matrices and rectangular generalized Vandermonde matrices of the block Toeplitz type.

Solutions of linear systems of equations are also presented in, e.g., [1], [2], and [4], where the coefficient matrices are Toeplitz, Hankel, Hilbert-type, Cauchy, and Vandermonde-type matrices.

The paper is organized as follows. In section 2, a right-inverse representation of the coefficient matrices $\mathcal{K}_\nu(z)$ and $\mathcal{M}_\tau(\rho)$ is introduced. In sections 3 and 4, a corresponding algorithm for the kernel of the coefficient matrices $\mathcal{K}_\nu(z)$ and $\mathcal{M}_\tau(\rho)$ is developed for the case $q = \nu + 1$, respectively, $p = \tau + 1$. The main conclusions are formulated in section 5. An algorithm for the kernel of $\mathcal{K}_\nu(z)$, when $q > \nu + 1$, is displayed in section 6.

## 2. A right-inverse: Case $q = \nu + 1$.

A right-inverse of $\mathcal{K}_\nu(z)$ is given for $q = \nu + 1$, which is a special form of the right-inverse presented in [5]. We introduce the $q \times q$ generalized Vandermonde matrix $\mathcal{T}_\nu^q(z)$ where

$$\mathcal{T}_\nu^q(z) = \left(\mathcal{T}_\nu^{(\nu)}(z), \mathcal{T}_\nu^{(\nu-1)}(z), \ldots, \mathcal{T}_\nu^{(0)}(z)\right)$$

and

$$\mathcal{T}_\nu^{(\nu-k)}(z) = \frac{\partial^{\nu-k}}{\partial z^{\nu-k}} u_q(z), \qquad k = 0, 1, \ldots, \nu.$$

The following lemma can now be formulated.

Lemma 2.1. *When $q = \nu + 1$ the relations*

$$\mathcal{K}_\nu(z)\,(I_q \otimes e_q) = \mathcal{T}_\nu^q(z),$$
$$\mathcal{K}_\nu(z)\left(\left(\mathcal{T}_\nu^q(z)\right)^{-1} \otimes e_q\right) = I_q$$

*hold true. Clearly, an appropriate right-inverse is then $(\mathcal{K}_\nu(z))_R^{-} = \left(\mathcal{T}_\nu^q(z)\right)^{-1} \otimes e_q$, where $e_q$ is the last standard basis vector in $\mathbb{R}^q$.*

*Proof.* Straightforward computation confirms the property. ☐

Consider the matrices $A$ and $B$ of size $m \times n$ and $p \times q$, respectively; then the $mp \times nq$ Kronecker product of the two matrices is defined as $A \otimes B = (a_{ij})B$ for all $i$ and $j$.

A choice for an appropriate right-inverse of $\mathcal{M}_\tau(\rho)$ when $p = \tau + 1$ is given in the following corollary.

COROLLARY 2.2. *When $p = \tau + 1$ a right-inverse of $\mathcal{M}_\tau(\rho)$ is given by*

$$\left(\mathcal{T}_\tau^p(\rho)\right)^{-1} \otimes e_p,$$

*where $e_p$ is the last standard basis vector in $\mathbb{R}^p$. We then have*

$$\mathcal{M}_\tau(\rho)\left(\left(\mathcal{T}_\tau^p(\rho)\right)^{-1} \otimes e_p\right) = I_p.$$

*Proof.* We have the property that the last column of $\operatorname{adj}(zI - C_p)$ is $u_p(z)$; this can be shown by equality (4.4), and this coincides with the last column of the matrix $u_p(z)u_p^{*\top}(z)$. This implies equality of the last column of the blocks composing $\mathcal{K}_\nu(z)$ and $\mathcal{M}_\tau(z)$. Since the construction of the right-inverse displayed in Lemma 2.1 is based on the last column of the blocks in $\mathcal{K}_\nu(z)$, the right-inverse set forth in Lemma 2.1 then also holds for $\mathcal{M}_\tau(z)$. □

In the next section an algorithm for the null space Ker $(\mathcal{K}_\nu(z))$ is displayed.

## 3. Ker $(\mathcal{K}_\nu(z))$ for the case $\nu + 1 = q$.

We shall specify the dimension of the null space Ker $(\mathcal{K}_\nu(z))$ in the next proposition.

PROPOSITION 3.1. *The null space Ker$(\mathcal{K}_\nu(z))$ has dimension equal to $q\nu$ and the rank of the coefficient matrix $\mathcal{K}_\nu(z)$ is $q$, when $\nu + 1 = q$.*

*Proof.* In Lemma 2.1, a right-inverse of the coefficient matrix $\mathcal{K}_\nu(z)$ is set forth. This implies that the $q \times q(\nu + 1)$ coefficient matrix $\mathcal{K}_\nu(z)$ is surjective or has full row rank; its rank is then $q$. By virtue of the dimension rule it can be concluded that dim Ker $(\mathcal{K}_\nu(z)) = q\nu$. □

We are going to prove that a basis of the null space Ker$(\mathcal{K}_\nu(z))$ is formed by the columns of the matrix

$$\mathcal{N} = \begin{pmatrix} \mathcal{U}(z) \\ J_{q\nu} \end{pmatrix}, \tag{3.1}$$

where $J_{q\nu}$ is the $q\nu$ rotation matrix

$$\begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{pmatrix},$$

and where the $q \times q\nu$ matrix $\mathcal{U}(z)$ will be specified later on.

Observe that $\mathcal{N}$ has full rank $q\nu$ since $J_{q\nu}$ is a nonsingular submatrix of $\mathcal{N}$. Therefore the columns of $\mathcal{N}$ form a basis of Ker$(\mathcal{K}_\nu(z))$.

The matrix $\mathcal{U}(z)$ is represented in the following form:

$$\mathcal{U}(z) = \frac{1}{\nu!}\left(\mathcal{U}_0(z), \mathcal{U}_1(z), \mathcal{U}_2(z), \dots, \mathcal{U}_{\nu-1}(z)\right). \tag{3.2}$$

The submatrices constituting (3.2) shall be specified in the next sections.

**3.1. A representation of $\mathcal{U}_0(z)$.** In Lemma 3.2 we prove that a column of the matrix $\mathcal{U}_0(z)$ which has the form

$$\mathcal{U}_0(z) = (\xi \odot u_q(z)) \otimes u_q^\top(z)), \tag{3.3}$$

and where the vector $\xi$ is given by

$$\xi = (\xi_i)\,,\ \xi_i = \left((-1)^i \binom{\nu}{i-1}\right)_{i=1,\ldots,\nu+1}, \tag{3.4}$$

belongs to the null space $\mathrm{Ker}(\mathcal{K}_\nu(z))$. The Hadamard product $\odot$ is defined by $A \odot B = (a_{ij}b_{ij})$ for $A = (a_{ij})$ and $B = (b_{ij})$ which are matrices of the same size.

Recall that the $m$th row of $\mathcal{K}_\nu(z)$ is given by

$$\frac{d^\nu}{dz^\nu}\left(z^{q-2+m}, z^{q-3+m}, \ldots, z^{m-1}\right) = \left(z^{q+m-i-\nu-1} \prod_{j=1}^{v} (n-i-j)\right)_{i=1,\ldots,q}. \tag{3.5}$$

We have the following lemma.

Lemma 3.2. *The $q\,(\nu+1)$ column vector composed of an arbitrary column of $\mathcal{U}_0(z)$ and the corresponding standard basis vector in $\mathbb{R}^{q\nu}$ belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$.*

*Proof.* The $k$th column of $\mathcal{U}_0(z)$ has elements

$$\frac{1}{\nu!}\left((-1)^i z^{k+i-2}\binom{\nu}{i-1}\right), \quad i = 1, \ldots, q. \tag{3.6}$$

The scalar product of (3.5) and (3.6) provides a monomial in $z^{n-\nu+k-3}$, where $n = q+m$, whose coefficient is given by

$$\begin{aligned} &\frac{1}{\nu!}\sum_{i=0}^{\nu}(-1)^{i+1}\binom{\nu}{i}(n-2-i)(n-3-i)\cdots(n-\nu-1-i) \\ &= \frac{1}{\nu!}\left\{\frac{d^\nu}{dx^\nu}\sum_{i=0}^{\nu}(-1)^{i+1}\binom{\nu}{i}x^{n-2-i}\right\}_{x=1} = -\frac{1}{\nu!}\left\{\frac{d^\nu}{dx^\nu}\left(x^{n-2-\nu}(x-1)^\nu\right)\right\}_{x=1}. \end{aligned} \tag{3.7}$$

The application of the Leibnitz rule to $\nu$-fold differentiation of a product of two functions yields the value $-\frac{1}{\nu!}\{x^{n-2-\nu}\nu!\}_{x=1} = -1$. Consequently, the scalar product of (3.5) and (3.6) is $-z^{n-\nu+k-3}$. This should be added to the product of the appropriate $z$-variable in the coefficient matrix $\mathcal{K}_\nu(z)$ by the nonzero element of the standard basis vector in the rotation matrix $J_{q\nu}$ which is $z^{n-\nu+k-3}$, so the sum is null. This completes the proof. ☐

**3.1.1. Summary of the construction of $\mathcal{U}_0(z)$.** *Step* 1. Introduce the vector $\xi$ according to (3.4).

*Step* 2. Define the columns of $\mathcal{U}_0(\sigma)$ according to (3.3).

**3.2. A representation of $\mathcal{U}_j(z)$ when $j = 1, 2, \ldots, \nu-1$.** We shall now describe the form of the matrices $\mathcal{U}_1(z), \mathcal{U}_2(z), \ldots, \mathcal{U}_{\nu-1}(z)$ that consist of the following structural representation:

$$\mathcal{U}_j(\sigma) = \left(\mathcal{U}_j^{(1)}(z)\;\mathcal{U}_j^{(2)}(z)\right),$$

for $j = 1, 2, \ldots, \nu-1$.

**3.2.1. A representation of $\boldsymbol{\mathcal{U}_j^{(1)}(z)}$.** In this section the matrix $\mathcal{U}_j^{(1)}(z)$ is displayed. It is Hankel type with the following configuration:

$$\mathcal{U}_j^{(1)}(z) = \left(\delta_j^1(z)\delta_j^2(z)\cdots\delta_j^{j+1}(z)\right), \tag{3.8}$$

where the $(\nu+1)$ basis column vector $\delta_j^{\ell+1}(z)$ has components

$$\left[\delta_j^{\ell+1}(z)\right]_i = \begin{cases} 0 & \text{for } i \leq j-\ell \text{ or } \nu+1-i \leq \ell \\ -j!(-z)^{i+\ell-j-1}\binom{\nu-j}{i+\ell-j-1} & \text{otherwise} \end{cases} \tag{3.9}$$

for $\ell = 0, 1, \ldots, j$. The following lemma is proved.

LEMMA 3.3. *The $q(\nu+1)$ column vector composed of any arbitrary column of $\mathcal{U}_j^{(1)}(z)$ and the corresponding standard basis vector in $\mathbb{R}^{q\nu}$ belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$.*

*Proof.* Set $j = p$ and $\ell = g$ in (3.9). As can be seen from (3.5), the appropriate nonzero elements of the scalar product of (3.9) with (3.5) provide a monomial in $z^{f-p-2-\nu}$, where $f = q+m+g$. Its coefficient is given by

(3.10)

$$\begin{aligned}
&\frac{p!}{\nu!}\sum_{i=0}^{\nu-p}(-1)^{i+1}\binom{\nu-p}{i}(f-p-2-i)(f-p-3-i)\cdots(f-p-1-i-\nu)\\
&\quad= \frac{p!}{\nu!}\left\{\frac{d^\nu}{dx^\nu}\sum_{i=0}^{\nu-p}(-1)^{i+1}\binom{\nu-p}{i}x^{f-p-2-i}\right\}_{x=1}\\
&\quad= -\frac{p!}{\nu!}\left\{\frac{d^\nu}{dx^\nu}\left[x^{f-2-\nu}\sum_{i=0}^{\nu-p}(-1)^{i}\binom{\nu-p}{i}x^{\nu-p-i}\right]\right\}_{x=1}\\
&\quad= -\frac{p!}{\nu!}\left\{\frac{d^\nu}{dx^\nu}x^{f-2-\nu}(x-1)^{\nu-p}\right\}_{x=1}\\
&\quad= -\frac{p!}{\nu!}\left\{0+0+\cdots+\binom{\nu}{\nu-p}\frac{d^p}{dx^p}x^{f-2-\nu}\frac{d^{\nu-p}}{dx^{\nu-p}}(x-1)^{\nu-p}+0+\cdots+0\right\}_{x=1}\\
&\quad= -(f-2-\nu)(f-3-\nu)\cdots(f-p-1-\nu).
\end{aligned}$$

The scalar product is then given by $-(f-2-\nu)(f-3-\nu)\cdots(f-p-1-\nu)z^{f-p-2-\nu}$. The appropriate element of the $m$th row of the coefficient matrix $\mathcal{K}_\nu(z)$ that is multiplied by the nonzero element of the corresponding standard basis vector in the rotation matrix $J_{q\nu}$ is $z^{q-w+m-1}$, where $w = \nu-g+1$, and the appropriate derivative is $p$. We therefore have

$$(d^p/dz^p)\,z^{f-\nu-2} = (f-2-\nu)(f-3-\nu)\cdots(f-p-1-\nu)\,z^{f-p-2-\nu}. \tag{3.11}$$

Adding (3.10) to (3.11) confirms that the $q(\nu+1)$ column vector composed of $\delta_j^{\ell+1}(z)$ given in (3.9) and the corresponding standard basis vector in the rotation matrix $J_{q\nu}$ belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$. This completes the proof. ☐

**3.2.2. Summary of construction of the matrix $\boldsymbol{\mathcal{U}_j^{(1)}(z)}$.** *Step* 1. Define vector $\delta_j^{\ell+1}(z)$ according to (3.9) for $\ell = 0, 1, \ldots, j$.

*Step* 2. Derive the columns of matrix $\mathcal{U}_j^{(1)}(z)$ according to (3.8).

**3.2.3. A representation of $\mathcal{U}_j^{(2)}(z)$.** For $j = 1, 2, 3, \ldots, \nu - 1$, the submatrix $\mathcal{U}_j^{(2)}(z)$ admits the structure

$$\mathcal{U}_j^{(2)}(z) = \left(\kappa_j^1(z)\kappa_j^2(z)\cdots\kappa_j^{\nu-j}(z)\right). \tag{3.12}$$

To specify the basis vectors $\kappa_j^1(z)\ \kappa_j^2(z)\ldots\kappa_j^{\nu-j}(z)$, we first compute recursively for $j = 1$ and $k = 2, 3, \ldots, \nu - j$ the appropriate column vectors according to

$$\kappa_1^k = \kappa_1^{k-1} + \xi, \tag{3.13}$$

where $\xi$ is given in (3.4). A solution to recursion ( 3.13) in terms of the initial vector $\kappa_j^1$ whose form shall be introduced below is

$$\kappa_1^k = \kappa_1^1 + (k-1)\xi. \tag{3.14}$$

We can now compute recursively for $j = 2, 3, \ldots, \nu - 1$, according to

$$\kappa_j^k = \kappa_j^{k-1} + j\kappa_{j-1}^k. \tag{3.15}$$

In the next proposition, an explicit solution to recursion equations (3.15) and (3.13) shall be displayed for $j = 1, 2, 3, \ldots, \nu - 1$.

PROPOSITION 3.4. *An explicit solution to the recursion equations* (3.15) *and* (3.13)*, expressed in terms of the initial vectors* $\kappa_j^1$, $\kappa_{j-1}^1, \ldots, \kappa_2^1, \kappa_1^1$ *and the known vector* $\xi$*, is given by*

$$\kappa_j^k = \sum_{i=0}^{j-1} i! \binom{j}{i} \binom{k-2+i}{k-2} \kappa_{j-i}^1 + j! \binom{k+j-2}{k-2} \xi. \tag{3.16}$$

*Proof.* The proof consists of using the recursion equations (3.15) and (3.14). Take $j = 2$, a combination of (3.15) and (3.14) yields for $k = 2, 3, 4, \ldots$

$$\begin{aligned}
\kappa_2^2 &= \kappa_2^1 + 2\kappa_1^1 + 2\xi\\
\kappa_2^3 &= \kappa_2^1 + 4\kappa_1^1 + 6\xi\\
\kappa_2^4 &= \kappa_2^1 + 6\kappa_1^1 + 12\xi\\
&\vdots\\
\kappa_2^k &= \kappa_2^1 + 2(k-1)\kappa_1^1 + k(k-1)\xi. \qquad (3.17)
\end{aligned}$$

Similarily when $j = 3, 4$, the recursion exercise yields for the $k$th column

$$\kappa_3^k = \kappa_3^1 + 3(k-1)\kappa_2^1 + 3k(k-1)\kappa_1^1 + k(k^2-1)\xi, \tag{3.18}$$

$$\kappa_4^k = \kappa_4^1 + 4(k-1)\kappa_3^1 + 6k(k-1)\kappa_2^1 + 4k(k^2-1)\kappa_1^1 + k(k^2-1)(k+2)\xi. \tag{3.19}$$

From (3.17), (3.18), and (3.19) can be concluded that for all values of $j$, the solution is then given by (3.16), where the case $j = 1$ is also included. When $j = 1$, (3.16) becomes (3.14). $\square$

The columns $\kappa_j^k$ for $k = 1, 2, \ldots, \nu - j$ and $j = 1, 2, 3, \ldots, \nu - 1$ are essential for displaying the corresponding columns of the submatrix $\mathcal{U}_j^{(2)}(z)$ set forth in (3.12) and to obtain

$$\kappa_j^k(z) = \kappa_j^k \odot z^k u_{\nu+1}(z). \tag{3.20}$$

In order to start the recursions, the $(\nu+1)$ initial column vector $\kappa_j^1$ shall be introduced. For $j = 1, 2, \dots, \nu-1$, the components of the vector $\kappa_j^1$ are given by

$$\left\{\begin{array}{c} \left[\kappa_j^1\right]_1 = (j+1)!,\ \left[\kappa_j^1\right]_2 = ((j+1)!/2)\,(2\nu - j)\,, \\ \left[\kappa_j^1\right]_i = j!\binom{\nu+1}{i} - s_i,\ i = 3, \dots, \nu - j, \\ \left[\kappa_j^1\right]_i = j!\binom{\nu+1}{i},\ i = \nu - j + 1, \dots, \nu + 1, \end{array}\right. \tag{3.21}$$

where the terms $s_\ell$, encountered if $\nu \geq 5$, are defined by

$$s_\ell = \begin{cases} j!\begin{pmatrix} \nu - j \\ \ell \end{pmatrix}, & \ell = 3, 4, \dots, \nu - j \quad \text{for } j = 1, 2, \dots, \nu - 3, \\ 0, & \qquad \alpha > \nu - 3 \qquad\quad \text{for } \kappa_\alpha^1. \end{cases} \tag{3.22}$$

From (3.22) it can be concluded that when $j = \nu - 2$ and $j = \nu - 1$, $s_\ell = 0$ for the corresponding initial vectors $\kappa_{\nu-2}^1$ and $\kappa_{\nu-1}^1$ of the submatrices $\mathcal{U}_{j=\nu-2}^{(2)}(z)$ and $\mathcal{U}_{j=\nu-1}^{(2)}(z)$, respectively. For the case $q \leq 5$, the initial vectors $\kappa_j^1$ do not contain the terms $s_\ell$ so the elements of $\kappa_j^1$ to be considered are the two first elements and then pursuing the reading upwards, starting from the last term at the bottom.

The first part of the right-hand side of (3.16) is displayed in order to better understand the development of the proof of Lemma 3.6 by setting $\vartheta = \sum_{i=0}^{j-1}\binom{k-2+i}{k-2}$,

$$j!\begin{pmatrix} \sum_{i=0}^{j-1}\binom{k-2+i}{k-2}(j-i+1) \\ \sum_{i=0}^{j-1}\binom{k-2+i}{k-2}\,((j-i+1)/2)\,(2\nu - j + i) \\ \vartheta\binom{\nu+1}{3} - \sum_{i=0}^{j-1}\binom{k-2+i}{k-2}\binom{\nu-j+i}{3} \\ \vartheta\binom{\nu+1}{4} - \sum_{i=0}^{j-1}\binom{k-2+i}{k-2}\binom{\nu-j+i}{4} \\ \vdots \\ \vartheta\binom{\nu+1}{\nu-j} - \sum_{i=0}^{j-1}\binom{k-2+i}{k-2}\binom{\nu-j+i}{\nu-j} \\ \vartheta\binom{\nu+1}{\nu-j+1} \\ \vdots \\ \vartheta\binom{\nu+1}{\nu+1} \end{pmatrix}. \tag{3.23}$$

The sign pattern of the elements in each column of $\mathcal{U}_j^{(2)}(z)$ is given by $(-1)^\ell$ with $\ell = 1, 2, \dots, \nu + 1$.

First some results which shall be used in the proof of Lemma 3.6 are set forth.

PROPOSITION 3.5. *The following equalities hold true*:

$$\sum_{i=0}^{j-1}\begin{pmatrix} k-2+i \\ k-2 \end{pmatrix} = \begin{pmatrix} k-2+j \\ k-1 \end{pmatrix}, \tag{3.24}$$

$$\sum_{i=1}^{j-1}\begin{pmatrix} k-2+i \\ k-2 \end{pmatrix} i = \frac{j(j-1)}{k}\begin{pmatrix} k-2+j \\ k-2 \end{pmatrix}, \tag{3.25}$$

$$\sum_{i=1}^{j-1}\begin{pmatrix} k-2+i \\ k-2 \end{pmatrix} i^2 = \frac{j(j-1)(1+(j-1)k)}{k(k+1)}\begin{pmatrix} k-2+j \\ k-2 \end{pmatrix}. \tag{3.26}$$

*Proof.* We shall prove the equalities (3.24), (3.25), and (3.26) by applying mathematical induction.

It is straightforward to see that the left-hand side of equality (3.24) yields the right-hand side when $j = 1$.

Assume for $j = p$ that

$$\sum_{i=0}^{p-1} \binom{k-2+i}{k-2} = \binom{k-2+p}{k-1}.$$

This implies that for $j = p + 1$,

$$\begin{aligned}\sum_{i=0}^{p} \binom{k-2+i}{k-2} &= \sum_{i=0}^{p-1} \binom{k-2+i}{k-2} + \binom{k-2+p}{k-2}\\ &= \binom{k-2+p}{k-1} + \binom{k-2+p}{k-2}\\ &= \binom{k-1+p}{k-1}.\end{aligned}$$

The last equality is based on the elementary identity for integers $n$ and $j$:

$$\binom{n}{j} + \binom{n}{j+1} = \binom{n+1}{j+1}. \tag{3.27}$$

The proof of (3.24) is completed. When $j = 2$, the left-hand side of equality (3.25) is $\binom{k-1}{k-2}$ and equals the right-hand side which becomes $\frac{2}{k}\binom{k}{k-2}$ Assume that (3.25) is true for $j = p$. Then

$$\sum_{i=0}^{p-1} \binom{k-2+i}{k-2} i = \frac{p(p-1)}{k} \binom{k-2+p}{k-2},$$

for $j = p + 1$,

$$\begin{aligned}\sum_{i=0}^{p} \binom{k-2+i}{k-2} i &= \sum_{i=0}^{p-1} \binom{k-2+i}{k-2} i + p\binom{k-2+p}{k-2}\\ &= \frac{p(p-1)}{k} \binom{k-2+p}{k-2} + p\binom{k-2+p}{k-2}\\ &= \frac{(k+p-1)!}{(k-2)!(p-1)!k} = \frac{p(p+1)}{k}\binom{k-1+p}{k-2}.\end{aligned}$$

This confirms (3.25). Finally we prove (3.26). When $j = 2$, the left-hand side of equality (3.26) is $\binom{k-1}{k-2}$ and equals the right-hand side which becomes $\frac{2}{k}\binom{k}{k-2}$ Assume for $j = p$ that

$$\sum_{i=0}^{p-1} \binom{k-2+i}{k-2} i^2 = \frac{p(p-1)(1+(p-1)k)}{k(k+1)} \binom{k-2+p}{k-2}.$$

This implies that for $j = p+1$,

$$\begin{aligned}\sum_{i=0}^{p}\binom{k-2+i}{k-2}i^2 &= \sum_{i=0}^{p-1}\binom{k-2+i}{k-2}i^2 + p^2\binom{k-2+p}{k-2}\\ &= \frac{p(p-1)(1+(p-1)k)}{k(k+1)}\binom{k-2+p}{k-2} + p^2\binom{k-2+p}{k-2}\\ &= \frac{(1+pk)\,(k+p-1)(k+p-2)!}{(k-2)!\,(p-1)!k(k+1)}\\ &= \frac{p(p+1)(1+pk)}{k(k+1)}\binom{k-1+p}{k-2}.\end{aligned}$$

This completes the proof. □

We shall now continue with the following lemma.

LEMMA 3.6. *The* $q\,(\nu+1)$ *column vector composed of* $\kappa_j^k(z)$*, described in* (3.16) *and* (3.20)*, and the corresponding standard basis vector in* $\mathbb{R}^{q\nu}$ *belongs to the null space of the coefficient matrix* $\mathcal{K}_\nu(z)$*.*

*Proof.* The scalar product of (3.20) and (3.5) provides a monomial in $z^{n+k-\nu-2}$. The $z$-variables will be reintroduced at a later stage for typographical brevity. The scalar product is first computed for the last $\nu-1$ entries of the first column of (3.23), then sets $j=p$ in (3.16) and takes (3.24) into consideration yielding

(3.28)

$$\begin{aligned}&\frac{p!}{\nu!}\binom{k-2+p}{k-1}\sum_{i=0}^{\nu-2}(-1)^{i+1}\binom{\nu+1}{3+i}(n-4-i)(n-5-i)\cdots(n-\nu-3-i)\\ &\quad= \frac{p!}{\nu!}\binom{k-2+p}{k-1}\left\{\frac{d^\nu}{dx^\nu}\sum_{i=0}^{\nu-2}(-1)^{i+1}\binom{\nu+1}{3+i}x^{n-4-i}\right\}_{x=1}\\ &\quad= -\frac{p!}{\nu!}\binom{k-2+p}{k-1}\left\{\frac{d^\nu}{dx^\nu}\left[x^{n-2-\nu}\sum_{i=0}^{\nu-2}(-1)^{i}\binom{\nu+1}{3+i}x^{\nu-2-i}\right]\right\}_{x=1}.\end{aligned}$$

Then set $j = 3+i$

$$= \frac{p!}{\nu!}\binom{k-2+p}{k-1}\left\{\frac{d^\nu}{dx^\nu}\left[x^{n-2-\nu}\left(\sum_{j=3}^{\nu+1}(-1)^{j}\binom{\nu+1}{j}x^{\nu+1-j}\right)\right]\right\}_{x=1}. \tag{3.29}$$

The following holds:

$$\sum_{j=3}^{\nu+1}(-1)^{j}\binom{\nu+1}{j}x^{\nu+1-j} = \sum_{j=0}^{\nu+1}(-1)^{j}\binom{\nu+1}{j}x^{\nu+1-j} - \sum_{j=0}^{2}(-1)^{j}\binom{\nu+1}{j}x^{\nu+1-j}.$$

Equation (3.29) becomes

(3.30)

$$\begin{aligned}&\frac{p!}{\nu!}\binom{k-2+p}{k-1}\frac{d^\nu}{dx^\nu}\left\{x^{n-2-\nu}\left((x-1)^{\nu+1} - x^{\nu+1} + (\nu+1)x^\nu - \frac{\nu(\nu+1)}{2}x^{\nu-1}\right)\right\}_{x=1}\\ &\quad= \frac{p!}{\nu!}\binom{k-2+p}{k-1}\left\{\begin{array}{c} -(n-1)(n-2)\cdots(n-\nu)\\ +(\nu+1)(n-2)(n-3)\cdots(n-\nu-1)\\ -\dfrac{\nu(\nu+1)}{2}(n-3)(n-4)\cdots(n-\nu-2)\end{array}\right\}.\end{aligned}$$

We shall now focus on the part of (3.23) that contains $s_\ell$. For that purpose an explicit representation is displayed,

$$p!\left\{\binom{k-2}{k-2}\begin{pmatrix}\binom{\nu-p}{3}\\ \binom{\nu-p}{4}\\ \vdots\\ \binom{\nu-p}{\nu-p}\end{pmatrix}+\binom{k-1}{k-2}\begin{pmatrix}\binom{\nu-p+1}{3}\\ \binom{\nu-p+1}{4}\\ \vdots\\ \binom{\nu-p+1}{\nu-p+1}\end{pmatrix}\right.$$

$$\left.+\cdots+\binom{k+p-3}{k-2}\begin{pmatrix}\binom{\nu-1}{3}\\ \binom{\nu-1}{4}\\ \vdots\\ \binom{\nu-1}{\nu-1}\end{pmatrix}\right\}.$$

The scalar product of (3.5) with each of the columns above can be expressed as follows, consider the index $\ell = 0, 1, 2, \ldots, p-1$, to obtain

$$\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\sum_{i=0}^{\nu-p+\ell-3}(-1)^i\binom{\nu-p+\ell}{3+i}(n-4-i)(n-5-i)\cdots(n-\nu-3-i)$$

$$=\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\left\{\frac{d^\nu}{dx^\nu}\sum_{i=0}^{\nu-p+\ell-3}(-1)^i\binom{\nu-p+\ell}{3+i}x^{n-4-i}\right\}_{x=1}.$$

Set $j = 3+i$; it then yields

$$\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\left\{\frac{d^\nu}{dx^\nu}\sum_{j=3}^{\nu-p+\ell}(-1)^{j-3}\binom{\nu-p+\ell}{j}x^{n-j-1}\right\}_{x=1}$$

$$=-\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\left\{\frac{d^\nu}{dx^\nu}x^{n-\nu+p-\ell-1}\left(\sum_{j=3}^{\nu-p+\ell}(-1)^j\binom{\nu-p+\ell}{j}x^{\nu-p+\ell-j}\right)\right\}_{x=1}$$

$$=-\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\left\{\frac{d^\nu}{dx^\nu}x^{n-\nu+p-\ell-1}\left(\sum_{j=0}^{\nu-p+\ell}(-1)^j\binom{\nu-p+\ell}{j}x^{\nu-p+\ell-j}\right.\right.$$

$$\left.\left.-\sum_{j=0}^{2}(-1)^j\binom{\nu-p+\ell}{j}x^{\nu-p+\ell-j}\right)\right\}_{x=1}$$

$$=-\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\left\{\frac{d^\nu}{dx^\nu}\left(x^{n-\nu+p-\ell-1}(x-1)^{\nu-p+\ell}-x^{n-1}+(\nu-p+\ell)x^{n-2}\right.\right.$$

$$\left.\left.-\frac{(\nu-p+\ell-1)(\nu-p+\ell)}{2}x^{n-3}\right)\right\}_{x=1}.$$

The first term can be expanded according to Leibnitz rule for $\nu$-fold differentiation of a product of two functions,

$$\left\{0+0+\cdots+\binom{\nu}{\nu-p+\ell}\frac{d^{p-\ell}}{dx^{p-\ell}}x^{n-\nu+p-\ell-1}\frac{d^{\nu-p+\ell}}{dx^{\nu-p+\ell}}(x-1)^{\nu-p+\ell}+0+\cdots+0\right\}_{x=1}.$$

The result is then

$$\frac{p!}{\nu!}\binom{k+\ell-2}{k-2}\left\{\begin{array}{c}(n-1)(n-2)\cdots(n-\nu)\\ -(\nu-p+\ell)(n-2)(n-3)\cdots(n-\nu-1)\\ +\dfrac{(\nu-p+\ell-1)\,(\nu-p+\ell)}{2}(n-3)(n-4)\cdots(n-\nu-2)\end{array}\right\} \tag{3.31}$$

$$-\frac{p!}{(p-\ell)!}\binom{k+\ell-2}{k-2}(n-\nu+p-\ell-1)(n-\nu+p-\ell-2)\cdots(n-\nu). \tag{3.32}$$

Since $q=\nu+1$, the terms $(n-\nu),(n-\nu-1)$, and $(n-\nu-2)$ in (3.31) are positive and $(n-\nu-2)\geq 0$.

The terms involving $\frac{p!}{\nu!}(n-1)(n-2)\cdots(n-\nu)$ appearing in (3.30) and (3.31), the latter for $\ell=0,1,2,\ldots,p-1$, when added yield

$$\begin{aligned}&-\binom{k-2+p}{k-1}+\sum_{i=0}^{p-1}\binom{k-2+i}{k-2}\\ &=-\binom{k-2+p}{k-1}+\binom{k-2+p}{k-1}=0.\end{aligned}$$

The last equality is established by virtue of (3.24). A more explicit expression for the first term in (3.23) is now considered, with the corresponding minus sign. By virtue of (3.24) and (3.25) it can be seen that

$$\begin{aligned}-\frac{p!}{\nu!}\sum_{i=0}^{p-1}\binom{k-2+i}{k-2}(p-i+1)&=-\frac{p!}{\nu!}\left\{(p+1)\binom{k-2+p}{k-1}\right.\\ &\left.-\frac{p(p-1)}{k}\binom{k-2+p}{k-2}\right\}.\end{aligned}$$

In the scalar product, the first term of (3.23) is multiplied by $(n-2)(n-3)\cdots(n-\nu-1)$. Summing up all of the terms involving this product, which also appears in (3.30) and (3.31), yields

$$\begin{aligned}&\left\{-(p+1)\binom{k-2+p}{k-1}+\frac{p(p-1)}{k}\binom{k-2+p}{k-2}+\binom{k+p-2}{k-1}(\nu+1)\right.\\ &\left.\quad-\sum_{i=0}^{p-1}\binom{k-2+i}{k-2}(\nu-p+i)\right\}\\ &=-(p+1)\binom{k-2+p}{k-1}+\frac{p(p-1)}{k}\binom{k-2+p}{k-2}+\binom{k+p-2}{k-1}(\nu+1)\\ &\quad-(\nu-p)\binom{k+p-2}{k-1}-\frac{p(p-1)}{k}\binom{k-2+p}{k-2}=0.\end{aligned}$$

The last equality is established by virtue of (3.24) and (3.25).

We focus now on an explicit form of the second term of (3.23). By virtue of (3.24), (3.25), and (3.26) we obtain

(3.33)
$$\begin{aligned}&\frac{p!}{2}\sum_{i=0}^{p-1}\binom{k-2+i}{k-2}(p-i+1)\,(2\nu-p+i)\\&\quad=\frac{p!}{2}\left\{\begin{matrix}(p+1)\,(2\nu-p)\binom{k-2+p}{k-1}+(2p-2\nu+1)\dfrac{p(p-1)}{k}\binom{k-2+p}{k-2}\\ -\dfrac{p(p-1)(1+(p-1)k)}{k(k+1)}\binom{k-2+p}{k-2}\end{matrix}\right\}.\end{aligned}$$

In the scalar product, the term (3.33) is multiplied by $(n-3)(n-4)\cdots(n-\nu-2)$. Summing up all of the terms involving this product, without $(p!/\nu!)$, which also appears in (3.30) and (3.31), yields next to (3.33),

$$\left\{-\binom{k-2+p}{k-1}\frac{\nu(\nu+1)}{2}+\frac{1}{2}\sum_{i=0}^{p-1}\binom{k-2+i}{k-2}(\nu-p+i-1)\,(\nu-p+i)\right\}. \tag{3.34}$$

We now collect all of the terms involved to obtain

$$\begin{aligned}&\frac{(p+1)\,(2\nu-p)}{2}\binom{k-2+p}{k-1}+(2p-2\nu+1)\frac{p(p-1)}{2k}\binom{k-2+p}{k-2}\\&-\frac{p(p-1)(1+(p-1)k)}{2k(k+1)}\binom{k-2+p}{k-2}-\frac{\nu(\nu+1)}{2}\binom{k-2+p}{k-1}\\&+\frac{(\nu-p-1)\,(\nu-p)}{2}\binom{k-2+p}{k-1}+(2\nu-2p-1)\frac{p(p-1)}{2k}\binom{k-2+p}{k-2}\\&+\frac{p(p-1)(1+(p-1)k)}{2k(k+1)}\binom{k-2+p}{k-2}=0;\end{aligned}$$

as in the other cases, this result is obtained by using (3.24), (3.25), and (3.26).

Consequently, the remaining terms are now collected—it concerns the term involving $\xi$ in (3.16), the appropriate scalar product is by virtue of (3.7) $-p!\binom{k-2+p}{k-2}$, and the terms derived from (3.32), for $\ell=0,1,2,\dots,p-1$, to obtain

$$-\sum_{i=0}^{p-1}\frac{p!}{(p-i)!}\binom{k+i-2}{k-2}(n-\nu+p-i-1)(n-\nu+p-i-2)\cdots(n-\nu).$$

The remaining terms can be summarized according to

$$-p!\sum_{i=0}^{p-1}\binom{k+i-2}{k-2}\binom{n-\nu+p-i-1}{n-\nu-1}-p!\binom{k-2+p}{k-2}. \tag{3.35}$$

Concerning (3.35), the following property will be proved:

$$\sum_{i=0}^{p}\binom{k+i-2}{k-2}\binom{n-\nu+p-i-1}{n-\nu-1}=\binom{n-\nu+k+p-2}{n-\nu+k-2}. \tag{3.36}$$

For proving (3.36), we consider, for all nonnegative integers $l, p$ and $n \geq p$,

$$\binom{n+l+1}{p} = \sum_{i=0}^{p} \binom{l+i}{i} \binom{n-i}{p-i}. \tag{3.37}$$

The proof is based on (3.27), which we rewrite as

$$\binom{m}{j} + \binom{m}{j+1} = \binom{m+1}{j+1}. \tag{3.38}$$

First we prove the formula for $n = p$. In this case the identity (3.37) reduces to

$$\binom{l+p+1}{p} = \sum_{i=0}^{p} \binom{l+i}{i}.$$

We use induction w.r.t. the variable $p$. The case $p = 0$ is a triviality. Assume that (3.37) holds true for a certain value of $p$. Then

$$\sum_{i=0}^{p+1} \binom{l+i}{i} = \sum_{i=0}^{p} \binom{l+i}{i} + \binom{l+p+1}{p+1}.$$

The first term on the right-hand side is equal to $\binom{l+p+1}{p}$ by hypothesis. Then adding the second term gives $\binom{l+p+2}{p+1}$ by virtue of (3.38).

The rest of the proof is by induction w.r.t. the variable $n$, $n \geq p$, since we have settled the case $n = p$. Consider the right-hand side of (3.37) with $n + 1$ instead of $n$ and compute using the induction hypothesis two times and repeatedly the identity (3.38),

$$\begin{aligned} \sum_{i=0}^{p} \binom{l+i}{i} \binom{n+1-i}{p-i} &= \sum_{i=0}^{p} \binom{l+i}{i} \binom{n-i}{p-i} + \sum_{i=0}^{p-1} \binom{l+i}{i} \binom{n-i}{p-1-i} \\ &= \binom{n+l+1}{p} + \binom{n+l+1}{p-1} \\ &= \binom{n+l+2}{p}. \end{aligned}$$

From (3.35) and (3.36) it can be concluded that the scalar product is equal to

$$-(n-\nu+k+p-2)(n-\nu+k+p-3)\cdots(n-\nu+k-1)z^{n+k-\nu-2}. \tag{3.39}$$

The corresponding nonzero element of the standard basis vector in the rotation matrix $J_{q\nu}$ is multiplied by $z^{n-w-1}$ for $w = \nu - p + 1 - k$, and the appropriate derivative is

$$(d^p/dz^p)\, z^{n-\nu+k+p-2} = (n-\nu+k+p-2)(n-\nu+k+p-3)\cdots(n-\nu+k-1)z^{n+k-\nu-2}. \tag{3.40}$$

Adding (3.39) to (3.40) confirms that the $q\,(\nu+1)$ column vector composed of vector $\kappa_j^k(z)$, described in (3.16) and (3.20), and the corresponding standard basis vector in the rotation matrix $J_{q\nu}$ belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$ when $s_\ell \neq 0$ in (3.21).

We now proceed with the proof when $s_\ell = 0$ in (3.21), the cases $j = \nu - 1$ and $j = \nu - 2$ are therefore considered. The initial vector (3.21) for the former case is

$$\kappa^1_{\nu-1} = (\nu-1)! \left( \nu, \binom{\nu+1}{2}, \binom{\nu+1}{3}, \dots, \binom{\nu+1}{\nu+1} \right)^\top . \tag{3.41}$$

The scalar product involving the first $\nu+1$ elements is displayed, and the last $\nu$ entries of (3.41) are first considered to obtain

$$\frac{(\nu-1)!}{\nu!} \sum_{i=0}^{\nu-1} (-1)^i \binom{\nu+1}{2+i} (n-3-i)(n-4-i)\cdots(n-\nu-2-i).$$

The same approach as we used to derive (3.29) yields

$$\frac{(\nu-1)!}{\nu!} \left\{ \begin{matrix} -(n-1)(n-2)\cdots(n-\nu) \\ +(\nu+1)(n-2)(n-3)\cdots(n-\nu-1) \end{matrix} \right\}.$$

Adding the scalar product involving the first element of (3.41) and (3.5) yields

$$\begin{aligned} &\left\{ \begin{matrix} -(n-2)(n-3)\cdots(n-\nu-1) \\ +\dfrac{(\nu-1)!\,(\nu+1)}{\nu!}(n-2)(n-3)\cdots(n-\nu-1) \\ -\dfrac{(\nu-1)!}{\nu!}(n-1)(n-2)\cdots(n-\nu) \end{matrix} \right\} \\ &= -(n-2)(n-3)\cdots(n-\nu). \end{aligned}$$

This result is obtained through straightforward calculation. It can now be concluded that the scalar product is

$$-(n-2)(n-3)\cdots(n-\nu) z^{n-\nu-1}. \tag{3.42}$$

Note for the case under study, $k = 1$ (it concerns the initial vector $\kappa^1_{\nu-1}$). The corresponding nonzero element of the standard basis vector in the rotation matrix $J_{q\nu}$ is multiplied by $z^{n-w-1}$ for $w = 2 - k$, $w = \nu - p + 1 - k$ in the general case. The appropriate derivative is then

$$\left(d^{\nu-1}/dz^{\nu-1}\right) z^{n-2} = (n-2)(n-3)\cdots(n-\nu) z^{n-\nu-1}. \tag{3.43}$$

Adding (3.42) to (3.43) confirms that when in (3.21) $s_\ell = 0$ and $j = \nu-1$, the $q\,(\nu+1)$ column vector, composed of vector $\kappa^1_{\nu-1}$, given in (3.41), and the corresponding standard basis vector in the rotation matrix $J_{q\nu}$, belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$.

The case $j = \nu - 2$ is considered next. The initial vector (3.21) is then

$$\kappa^1_{\nu-2} = (\nu-2)! \left( (\nu-1), ((\nu-1)/2)(\nu+2), \binom{\nu+1}{3}, \dots, \binom{\nu+1}{\nu+1} \right)^\top . \tag{3.44}$$

The scalar product involving the first $\nu+1$ elements is displayed, and the last $(\nu-1)$ entries of (3.44) are first considered to obtain

$$\frac{(\nu-2)!}{\nu!} \sum_{i=0}^{\nu-2} (-1)^{i+1} \binom{\nu+1}{3+i} (n-4-i)(n-5-i)\cdots(n-\nu-3-i).$$

According to (3.28) we have

$$\frac{(\nu-2)!}{\nu!}\left\{\begin{array}{c} -(n-1)(n-2)\cdots(n-\nu) \\ +(\nu+1)(n-2)(n-3)\cdots(n-\nu-1) \\ -\dfrac{\nu(\nu+1)}{2}(n-3)(n-4)\cdots(n-\nu-2) \end{array}\right\}.$$

The scalar product involving the first and second elements of (3.44) and (3.5) are

$$\begin{array}{l} -\dfrac{(\nu-1)!}{\nu!}(n-2)(n-3)\cdots(n-\nu-1) \\ \text{and}\quad \dfrac{((\nu-1)!/2)\,(\nu+2)}{\nu!}(n-3)(n-4)\cdots(n-\nu-2), \end{array}$$

respectively. Summing all of the terms yields

$$\begin{array}{l} \left\{\begin{array}{c} \frac{(\nu-2)!(\nu+1)-(\upsilon-1)!}{\nu!}(n-2)(n-3)\cdots(n-\nu-1) \\ +\frac{(\nu-1)!(\nu+2)-(\nu-2)!\nu(\nu+1)}{2(\nu!)}(n-3)(n-4)\cdots(n-\nu-2) \\ -\frac{(\nu-2)!}{\nu!}(n-1)(n-2)\cdots(n-\nu) \end{array}\right\} \\ = -(n-3)(n-4)\cdots(n-\nu). \end{array}$$

This result is obtained through straightforward computation. It can now be concluded that the scalar product is

$$-(n-3)(n-4)\cdots(n-\nu)z^{n-\nu-1}. \tag{3.45}$$

Note for the case under study, $k=1$ (it concerns the initial vector $\kappa_{\nu-2}^1$).

The corresponding nonzero element of the standard basis vector in the rotation matrix $J_{q\nu}$ is multiplied by $z^{n-w-1}$ for $w=3-k$ and $w=\nu-p+1-k$ in the general case. The appropriate derivative is then

$$\left(d^{\nu-2}/dz^{\nu-2}\right)z^{n-3}=(n-3)(n-4)\cdots(n-\nu)z^{n-\nu-1}. \tag{3.46}$$

Adding (3.45) to (3.46) confirms that when in (3.21) $s_\ell=0$ and $j=\nu-2$, the $q\,(\nu+1)$ column vector, composed of vector $\kappa_{\nu-2}^1$, given in (3.44), and the corresponding standard basis vector in the rotation matrix $J_{q\nu}$, belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$.

It can be concluded that the $q\,(\nu+1)$ column vector, composed of vector $\kappa_j^k(z)$, described in (3.16) and (3.20), and the corresponding standard basis vector in the rotation matrix $J_{q\nu}$, belongs to the null space of the coefficient matrix $\mathcal{K}_\nu(z)$. The proof of Lemma 3.6 is now complete. ☐

**3.2.4. Summary of the construction of matrix $\mathcal{U}_j^{(2)}(z)$.** *Step* 1. Define the initial vectors $\kappa_j^1$ given in (3.21) for the values of $j=1,2,3,\ldots,\nu-1$.

*Step* 2. Expand (3.16) for the corresponding values of $j=1,2,3,\ldots,\nu-1$.

*Step* 3. Compute the columns of $\mathcal{U}_j^{(2)}(z)$ according to (3.20) for the corresponding values of $j=1,2,3,\ldots,\nu-1$.

In the next section an example will illustrate the results set forth in previous sections.

**3.3. Example Ker $(\mathcal{K}_\nu(z))$ for the case $\nu + 1 = 6$.** This case will be illustrated for $q = 6$ and $\nu = 5$. The first submatrix contained in the null space of $\mathcal{K}_\nu(z)$ is then

$$\mathcal{U}_0(z) = \begin{pmatrix} -1 & -z & -z^2 & -z^3 & -z^4 & -z^5 \\ 5z & 5z^2 & 5z^3 & 5z^4 & 5z^5 & 5z^6 \\ -10z^2 & -10z^3 & -10z^4 & -10z^5 & -10z^6 & -10z^7 \\ 10z^3 & 10z^4 & 10z^5 & 10z^6 & 10z^7 & 10z^8 \\ -5z^4 & -5z^5 & -5z^6 & -5z^7 & -5z^8 & -5z^9 \\ z^5 & z^6 & z^7 & z^8 & z^9 & z^{10} \end{pmatrix}.$$

This is followed by the second class of submatrices $\mathcal{U}_j(z)$ when $j = 1, 2, 3, 4$,

$$\mathcal{U}^{(1)}_{j=1}(z) = \begin{pmatrix} 0 & -1 \\ -1 & 4z \\ 4z & -6z^2 \\ -6z^2 & 4z^3 \\ 4z^3 & -z^4 \\ -z^4 & 0 \end{pmatrix}, \quad \mathcal{U}^{(1)}_{j=2}(z) = \begin{pmatrix} 0 & 0 & -2 \\ 0 & -2 & 6z \\ -2 & 6z & -6z^2 \\ 6z & -6z^2 & 2z^3 \\ -6z^2 & 2z^3 & 0 \\ 2z^3 & 0 & 0 \end{pmatrix},$$

$$\mathcal{U}^{(1)}_{j=3}(z) = \begin{pmatrix} 0 & 0 & 0 & -6 \\ 0 & 0 & -6 & 12z \\ 0 & -6 & 12z & -6z^2 \\ -6 & 12z & -6z^2 & 0 \\ 12z & -6z^2 & 0 & 0 \\ -6z^2 & 0 & 0 & 0 \end{pmatrix},$$

and

$$\mathcal{U}^{(1)}_{j=4}(z) = \begin{pmatrix} 0 & 0 & 0 & 0 & -24 \\ 0 & 0 & 0 & -24 & 24z \\ 0 & 0 & -24 & 24z & 0 \\ 0 & -24 & 24z & 0 & 0 \\ -24 & 24z & 0 & 0 & 0 \\ 24z & 0 & 0 & 0 & 0 \end{pmatrix}.$$

This is then followed by a class of submatrices $\mathcal{U}^{(2)}_j(z)$ when $j = 1, 2, 3, 4$,

$$\mathcal{U}^{(2)}_{j=1}(z) = \begin{pmatrix} -2z & -3z^2 & -4z^3 & -5z^4 \\ 9z^2 & 14z^3 & 19z^4 & 24z^5 \\ -16z^3 & -26z^4 & -36z^5 & -46z^6 \\ 14z^4 & 24z^5 & 34z^6 & 44z^7 \\ -6z^5 & -11z^6 & -16z^7 & -21z^8 \\ z^6 & 2z^7 & 3z^8 & 4z^9 \end{pmatrix},$$

$$\mathcal{U}^{(2)}_{j=2}(z) = \begin{pmatrix} -6z & -12z^2 & -20z^3 \\ 24z^2 & 52z^3 & 90z^4 \\ -38z^3 & -90z^4 & -162z^5 \\ 30z^4 & 78z^5 & 146z^6 \\ -12z^5 & -34z^6 & -66z^7 \\ 2z^6 & 6z^7 & 12z^8 \end{pmatrix},$$

$$\mathcal{U}^{(2)}_{j=3}(z) = \begin{pmatrix} -24z & -60z^2 \\ 84z^2 & 240z^3 \\ -120z^3 & -390z^4 \\ 90z^4 & 324z^5 \\ -36z^5 & -138z^6 \\ 6z^6 & 24z^7 \end{pmatrix}, \text{ and}$$

$$\mathcal{U}^{(2)}_{j=4}(z) = \begin{pmatrix} -120z \\ 360z^2 \\ -480z^3 \\ 360z^4 \\ -144z^5 \\ 24z^6 \end{pmatrix}.$$

Insertion of $\mathcal{U}_0(z)$ and the matrices

$$\begin{aligned} \mathcal{U}_1(z) &= \left( \mathcal{U}^{(1)}_{j=1}(z)\, \mathcal{U}^{(2)}_{j=1}(z) \right), \\ \mathcal{U}_2(z) &= \left( \mathcal{U}^{(1)}_{j=2}(z)\, \mathcal{U}^{(2)}_{j=2}(z) \right), \\ \mathcal{U}_3(z) &= \left( \mathcal{U}^{(1)}_{j=3}(z)\, \mathcal{U}^{(2)}_{j=3}(z) \right), \\ \mathcal{U}_4(z) &= \left( \mathcal{U}^{(1)}_{j=4}(z)\, \mathcal{U}^{(2)}_{j=4}(z) \right), \end{aligned}$$

in (3.2) yields the form

$$\mathcal{U}(z) = \frac{1}{5!} \left( \mathcal{U}_0(z), \mathcal{U}_1(z), \mathcal{U}_2(z), \mathcal{U}_3(z), \mathcal{U}_4(z) \right).$$

The columns that compose $\binom{\mathcal{U}(z)}{J_{30}}$ span Ker $(\mathcal{K}_\nu(z))$ when $q = 6$ and $\nu = 5$.

In the next section the null space of the coefficient matrix $\mathcal{M}_\tau(\rho)$ is set forth.

## 4. A representation of Ker $(\mathcal{M}_\tau(\rho))$.

In this section a representation of the subspace Ker$(\mathcal{M}_\tau(\rho))$ is displayed for the case $\tau + 1 = p$. The coefficient matrix $\mathcal{M}_\tau(z)$ is considered for $z = \rho$, and a motivation is formulated below. We shall first focus on the dimension of the null space Ker$(\mathcal{M}_\tau(\rho))$.

PROPOSITION 4.1. *The null space Ker*$(\mathcal{M}_\tau(\rho))$ *has dimension equal to* $p\tau$ *and the rank of the coefficient matrix* $(\mathcal{M}_\tau(\rho))$ *is* $p$, *when* $\tau + 1 = p$.

*Proof.* By virtue of Corollary 2.2, a similar argument as in Proposition 3.1 holds for the coefficient matrix $\mathcal{M}_\tau(\rho)$; see also Lemma 2.4 in [5]. It can be concluded that the $p \times p(\tau + 1)$ coefficient matrix $\mathcal{M}_\tau(\rho)$ is surjective or has full row rank; its rank is then $p$. By virtue of the dimension rule, it can be concluded that dim Ker$(\mathcal{K}_\nu(z)) = p\tau$. ☐

We can essentially reduce the problem of computing the null space Ker$(\mathcal{M}_\tau(\rho))$ to the computation of the kernel of the matrix $\mathcal{K}_\tau(\rho)$. The vectors contained in

$$\mathcal{G} = \begin{pmatrix} \mathcal{Y}(\rho) \\ J_{p\tau} \end{pmatrix} \tag{4.1}$$

span the null space of $\mathcal{M}_\tau(\rho)$, where $J_{p\tau}$ is the $p\tau$ rotation matrix.

Observe that $\mathcal{G}$ has full rank $p\tau$ since $J_{p\tau}$ is a nonsingular submatrix of $\mathcal{G}$. Therefore the columns of $\mathcal{G}$ form a basis of Ker$(\mathcal{M}_\tau(\rho))$.

Write

$$\mathcal{Y}(\rho) = \frac{1}{\tau!}\left(\mathcal{Y}_0(\rho), \mathcal{Y}_1(\rho), \mathcal{Y}_2(\rho), \ldots, \mathcal{Y}_{\tau-1}(\rho)\right), \tag{4.2}$$

where

$$\mathcal{Y}_0(\rho) = \mathcal{U}_0^\top(\rho)$$

and

$$\mathcal{Y}_j(\rho) = \begin{pmatrix} \mathcal{Y}_j^{(1)}(\rho) \\ \mathcal{Y}_j^{(2)}(\rho) \end{pmatrix} = \begin{pmatrix} \left(\mathcal{U}_j^{(1)}(\rho)\right)^\top \\ \left(\mathcal{U}_j^{(2)}(\rho)\right)^\top \end{pmatrix} = \mathcal{U}_j^\top(\rho) \quad \text{for} \quad j = 1, 2, \ldots, \tau - 1.$$

The matrices $\mathcal{U}_0(\rho)$, $\mathcal{U}_j^{(1)}(\rho)$, and $\mathcal{U}_j^{(2)}(\rho)$ are given in section 3.

In section 3.2 of [5], the vector $y \in \text{Ker }(\mathcal{M}_\tau(\rho))$ is computed according to

$$y = (I_{\tau+1} \otimes S(f))^{-1} x, \tag{4.3}$$

where $x \in \text{Ker }(\mathcal{K}_\tau(\rho))$ and the $p \times p$ symmetrizer $S(f)$ is associated with a polynomial $f(z)$ of degree $p$. Consider $f(z) = z^p + a_1 z^{p-1} + a_2 z^{p-2} + \cdots + a_p$, then the $p \times p$ matrix $S(f)$ is

$$S(f) = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ a_1 & 1 & 0 & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & 1 & 0 \\ a_{p-1} & & & a_1 & 1 \end{pmatrix}.$$

Formula (4.3) is derived from an equality which connects the matrices $\text{adj}(zI - C_p)$ and $u_p(z) u_p^{*\top}(z)$, where $u_p(z)$ and $u_p^*(z)$ are defined in (1.4). From [6], we take Proposition 3.1 which gives the identity

$$\text{adj}\,(zI - C_p) = u_p(z) a^\top(z)\, J_p - \pi(z) \sum_{i=0}^{p-1} z^i S^{i+1}. \tag{4.4}$$

The vector $a\,(z)$ is the $p$-vector $(a_0\,(z), \ldots, a_{p-1}\,(z))$, where $a_k\,(z)$ is the Hörner polynomial defined by $a_0\,(z) = 1$ and $a_k\,(z) = z a_{k-1}\,(z) + a_k$, and $a_k$ is an entry of $C_p$. Note that $a_p\,(z)$ is the characteristic polynomial of $C_p$. We further have that the rotation matrix $J_p \in \mathbb{R}^{p \times p}$, $\pi(z)$ is the characteristic polynomial of $C_p$ and $S$ denotes the shift matrix, so $S_{ij} = \delta_{i,j+1}$. Observe that the property $a^\top(z)\, J_p = u_p^*(z)^\top S(f)$ is used in (4.4) to obtain (4.3).

If $z = \rho$, where $\rho$ is an eigenvalue of the companion matrix $C_p$, then the second term in the right-hand side of (4.4) vanishes. It is then possible to derive form (4.3) (see [5]), and this is the reason why in this section one chooses working with $z = \rho$ instead of $z$.

A relation between the submatrices $\mathcal{Y}(\rho)$ in (4.2) and $\mathcal{U}(\rho)$ in (3.2) can now be displayed through equality (4.3). For that purpose we denote the vectors $v_0(\rho)$, $v_1(\rho)$, $v_2(\rho), \ldots, v_{\tau-1}(\rho)$ as being the first columns of the submatrices $\mathcal{U}_0(\rho)$, $\mathcal{U}_1(\rho)$,

$\mathcal{U}_2(\rho), \ldots, \mathcal{U}_{\tau-1}(\rho)$, given in (3.2). Whereas the vectors $w_0(\rho)$, $w_1(\rho)$, $w_2(\rho), \ldots, w_{\tau-1}(\rho)$ represent the first rows of the same submatrices. The following property is now summarized in the lemma.

LEMMA 4.2. *By virtue of* (4.3), *the following equalities hold true for* $i = 0, 1, 2, \ldots, \tau - 1$*:*

$$y_i(\rho) = S^{-1}(f)v_i(\rho) = w_i^\top(\rho),$$

*where* $y_0(\rho)$, $y_1(\rho)$, $y_2(\rho), \ldots, y_{\tau-1}(\rho)$ *are the first columns of the submatrices* $\mathcal{Y}_0(\rho)$, $\mathcal{Y}_1(\rho), \mathcal{Y}_2(\rho), \ldots, \mathcal{Y}_{\tau-1}(\rho)$ *given in* (4.2).

*Proof.* Straightforward matrix multiplications $S^{-1}(f)v_i(\rho)$ confirm the property. ☐

This leads to the main result of this section.

COROLLARY 4.3. *For the case* $\tau + 1 = p$, *the span of the null space of* $\mathcal{M}_\tau(\rho)$ *is*

$$\begin{pmatrix} \mathcal{Y}(\rho) \\ J_{p\tau} \end{pmatrix},$$

*where* $\mathcal{Y}(\rho)$ *is given by* (4.2).

*Proof.* It can be verified through matrix multiplications that

$$\mathcal{M}_\tau(\rho) \begin{pmatrix} \mathcal{Y}(\rho) \\ J_{p\tau} \end{pmatrix} = 0$$

holds. This is in agreement with the appropriate dimensions specified above. ☐

It can be seen from (4.3) that for every vector $y \in \text{Ker}\ (\mathcal{M}_\tau(\rho))$ computed according to the approach suggested in [5], the symmetrizer $S(f)$, a lower triangular and Toeplitz matrix has to be inverted once. However, this is combined with $p\tau$ matrix multiplications by the corresponding vector $x \in \text{Ker}\ (\mathcal{K}_\tau(\rho))$. This is in agreement with the dimension of the null space of $\mathcal{M}_\tau(\rho)$. In this paper there are neither matrix multiplications nor inversions involved in the construction of the span of the null spaces of $\mathcal{K}_\tau(\rho)$ and $\mathcal{M}_\tau(\rho)$. The null space of $\mathcal{M}_\tau(\rho)$ is obtained by transposing the submatrices contained in the null space of $\mathcal{K}_\tau(\rho)$. Consequently, when the algorithm of the null space of $\mathcal{K}_\tau(\rho)$ is available, the new approach does not require any computational exercise for displaying the span of the null space of $\mathcal{M}_\tau(\rho)$. In the next section an example of the null space of $\mathcal{M}_\tau(\rho)$ is set forth so that the property emphasized in this section will be illustrated.

**4.1. Example Ker ($\mathcal{M}_\tau(\rho)$) when $\tau + 1 = 7$.** This case will be illustrated for $p = 7$ and $\tau = 6$. The first matrix is then

$$\mathcal{Y}_0(\rho) = \begin{pmatrix} -1 & 6\rho & -15\rho^2 & 20\rho^3 & -15\rho^4 & 6\rho^5 & -\rho^6 \\ -\rho & 6\rho^2 & -15\rho^3 & 20\rho^4 & -15\rho^5 & 6\rho^6 & -\rho^7 \\ -\rho^2 & 6\rho^3 & -15\rho^4 & 20\rho^5 & -15\rho^6 & 6\rho^7 & -\rho^8 \\ -\rho^3 & 6\rho^4 & -15\rho^5 & 20\rho^6 & -15\rho^7 & 6\rho^8 & -\rho^9 \\ -\rho^4 & 6\rho^5 & -15\rho^6 & 20\rho^7 & -15\rho^8 & 6\rho^9 & -\rho^{10} \\ -\rho^5 & 6\rho^6 & -15\rho^7 & 20\rho^8 & -15\rho^9 & 6\rho^{10} & -\rho^{11} \\ -\rho^6 & 6\rho^7 & -15\rho^8 & 20\ \rho^9 & -15\rho^{10} & 6\rho^{11} & -\rho^{12} \end{pmatrix}.$$

The following class of matrices are for $j = 1, 2, 3, 4, 5$:

$$\mathcal{Y}_{j=1}^{(1)}(\rho) = \begin{pmatrix} 0 & -1 & 5\rho & -10\rho^2 & 10\rho^3 & -5\rho^4 & \rho^5 \\ -1 & 5\rho & -10\rho^2 & 10\rho^3 & -5\rho^4 & \rho^5 & 0 \end{pmatrix},$$

$$\mathcal{Y}^{(1)}_{j=2}(\rho) = \begin{pmatrix} 0 & 0 & -2 & 8\rho & -12\rho^2 & 8\rho^3 & -2\rho^4 \\ 0 & -2 & 8\rho & -12\rho^2 & 8\rho^3 & -2\ \rho^4 & 0 \\ -2 & 8\rho & -12\rho^2 & 8\rho^3 & -2\rho^4 & 0 & 0 \end{pmatrix} ,$$

$$\mathcal{Y}^{(1)}_{j=3}(\rho) = \begin{pmatrix} 0 & 0 & 0 & -6 & 18\rho & -18\rho^2 & 6\rho^3 \\ 0 & 0 & -6 & 18\rho & -18\rho^2 & 6\rho^3 & 0 \\ 0 & -6 & 18\rho & -18\rho^2 & 6\rho^3 & 0 & 0 \\ -6 & 18\rho & -18\rho^2 & 6\rho^3 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathcal{Y}^{(1)}_{j=4}(\rho) = \begin{pmatrix} 0 & 0 & 0 & 0 & -24 & 48\rho & -24\rho^2 \\ 0 & 0 & 0 & -24 & 48\rho & -24\rho^2 & 0 \\ 0 & 0 & -24 & 48\rho & -24\rho^2 & 0 & 0 \\ 0 & -24 & 48\rho & -24\rho^2 & 0 & 0 & 0 \\ -24 & 48\rho & -24\rho^2 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathcal{Y}^{(1)}_{j=5}(\rho) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -120 & 120\rho \\ 0 & 0 & 0 & 0 & -120 & 120\rho & 0 \\ 0 & 0 & 0 & -120 & 120\rho & 0 & 0 \\ 0 & 0 & -120 & 120\rho & 0 & 0 & 0 \\ 0 & -120 & 120\rho & 0 & 0 & 0 & 0 \\ -120 & 120\rho & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The matrices $\mathcal{Y}^{(2)}_j(\rho)$ with $j = 1, 2, 3, 4, 5$ are now displayed:

$$\mathcal{Y}^{(2)}_{j=1}(\rho) = \begin{pmatrix} -2\rho & 11\rho^2 & -25\rho^3 & 30\rho^4 & -20\rho^5 & 7\rho^6 & -\rho^7 \\ -3\rho^2 & 17\rho^3 & -40\rho^4 & 50\rho^5 & -35\rho^6 & 13\rho^7 & -2\rho^8 \\ -4\rho^3 & 23\rho^4 & -55\rho^5 & 70\rho^6 & -50\rho^7 & 19\rho^8 & -3\rho^9 \\ -5\rho^4 & 29\rho^5 & -70\rho^6 & 90\rho^7 & -65\rho^8 & 25\rho^9 & -4\rho^{10} \\ -6\rho^5 & 35\rho^6 & -85\rho^7 & 110\rho^8 & -80\rho^9 & 31\rho^{10} & -5\rho^{11} \end{pmatrix},$$

$$\mathcal{Y}^{(2)}_{j=2}(\rho) = \begin{pmatrix} -6\rho & 30\rho^2 & -62\rho^3 & 68\rho^4 & -42\rho^5 & 14\rho^6 & -2\rho^7 \\ -12\rho^2 & 64\rho^3 & -142\rho^4 & 168\rho^5 & -112\rho^6 & 40\rho^7 & -6\rho^8 \\ -20\rho^3 & 110\rho^4 & -252\rho^5 & 308\rho^6 & -212\rho^7 & 78\ \rho^8 & -12\rho^9 \\ -30\rho^4 & 168\rho^5 & -392\rho^6 & 488\rho^7 & -342\rho^8 & 128\ \rho^9 & -20\rho^{10} \end{pmatrix},$$

$$\mathcal{Y}^{(2)}_{j=3}(\rho) = \begin{pmatrix} -24\rho & 108\rho^2 & -204\rho^3 & 210\rho^4 & -126\rho^5 & 42\rho^6 & -6\rho^7 \\ -60\rho^2 & 300\rho^3 & -630\rho^4 & 714\rho^5 & -462\rho^6 & 162\ \rho^7 & -24\ \rho^8 \\ -120\rho^3 & 630\rho^4 & -1386\rho^5 & 1638\rho^6 & -1098\rho^7 & 396\ \rho^8 & -60\rho^9 \end{pmatrix},$$

$$\mathcal{Y}^{(2)}_{j=4}(\rho) = \begin{pmatrix} -120\rho & 480\rho^2 & -840\rho^3 & 840\rho^4 & -504\rho^5 & 168\rho^6 & -24\rho^7 \\ -360\rho^2 & 1680\rho^3 & -3360\rho^4 & 3696\rho^5 & -2352\rho^6 & 816\ \rho^7 & -120\rho^8 \end{pmatrix},$$

$$\mathcal{Y}^{(2)}_{j=5}(\rho) = \begin{pmatrix} -720\rho & 2520\rho^2 & -4200\rho^3 & 4200\rho^4 & -2520\rho^5 & 840\rho^6 & -120\rho^7 \end{pmatrix}.$$

Insertion of the matrix $\mathcal{Y}_0(\rho)$ in (4.2) followed by

$$\mathcal{Y}_1(\rho) = \begin{pmatrix} \mathcal{Y}^{(1)}_{j=1}(\rho) \\ \mathcal{Y}^{(2)}_{j=1}(\rho) \end{pmatrix}, \quad \mathcal{Y}_2(\rho) = \begin{pmatrix} \mathcal{Y}^{(1)}_{j=2}(\rho) \\ \mathcal{Y}^{(2)}_{j=2}(\rho) \end{pmatrix},$$

$$\mathcal{Y}_3(\rho) = \begin{pmatrix} \mathcal{Y}^{(1)}_{j=3}(\rho) \\ \mathcal{Y}^{(2)}_{j=3}(\rho) \end{pmatrix}, \quad \mathcal{Y}_4(\rho) = \begin{pmatrix} \mathcal{Y}^{(1)}_{j=4}(\rho) \\ \mathcal{Y}^{(2)}_{j=4}(\rho) \end{pmatrix}, \quad \mathcal{Y}_5(\rho) = \begin{pmatrix} \mathcal{Y}^{(1)}_{j=5}(\rho) \\ \mathcal{Y}^{(2)}_{j=5}(\rho) \end{pmatrix}$$

yields the representation

$$\mathcal{Y}(\rho) = \frac{1}{6!}\left( \mathcal{Y}_0(\rho), \mathcal{Y}_1(\rho), \mathcal{Y}_2(\rho), \mathcal{Y}_3(\rho), \mathcal{Y}_4(\rho), \mathcal{Y}_5(\rho)\right).$$

The vectors contained in $\binom{\mathcal{Y}(\rho)}{J_{42}}$ span the null space of $\mathcal{M}_\tau(\rho)$ when $p = 7$ and $\tau = 6$. It is straightforward to verify that when the matrices $\mathcal{Y}_0(\rho)$, $\mathcal{Y}_j^{(1)}(\rho)$, and $\mathcal{Y}_j^{(2)}(\rho)$, with $j = 1, 2, 3, 4, 5$, are transposed and inserted in (3.2) accordingly, one obtains the null space of $\mathcal{K}_\tau(\rho)$.

A summary of the results will be given in the next section.

**5. Main conclusions.** The results displayed in sections 2–4 allow us to present an explicit representation of the solutions to the linear systems of equations introduced in this paper. The solutions, (1.5) and (1.6), to the linear system of (1.1) and (1.2) are given by

$$\begin{aligned} X &= (\mathcal{K}_\nu(z))^- \,\mathcal{E} + \mathcal{W}(z) \text{ with } \mathcal{W}(z) \in \operatorname{Ker}(\mathcal{K}_\nu(z)),\\ Y &= (\mathcal{M}_\tau(\rho))^- \,\mathcal{R} + \mathcal{L}(\rho) \text{ with } \mathcal{L}(\rho) \in \operatorname{Ker}(\mathcal{M}_\tau(\rho)). \end{aligned}$$

An explicit expression for $(\mathcal{K}_\nu(z))^-$ and $\mathcal{W}(z)$ has been developed in sections 2 and 3, respectively, and a solution to the linear system of equations (1.1) is implementable. Analogously for the expressions $(\mathcal{M}_\tau(\rho))^-$ and $\mathcal{L}(\rho)$, constructed in sections 2 and 4, respectively, a solution to the linear system of (1.2) is implementable.

In the next section an algorithm for the null space $\operatorname{Ker}(\mathcal{K}_\nu(z))$, for the case $\nu + 1 < q$, is presented. It is a variant of the algorithm displayed in section 3.

**6. Ker ($\mathcal{K}_\nu(z)$) for the case $\nu + 1 < q$.** In this section the case $\nu + 1 < q$ is considered for the null space Ker ($\mathcal{K}_\nu(z)$). We then have $\operatorname{rank}(\mathcal{K}_\nu(z)) = \nu + 1$ so that $\dim \operatorname{Ker}(\mathcal{K}_\nu(z)) = (q - 1)(\nu + 1)$. In this case the coefficient matrix $\mathcal{K}_\nu(z)$ is not surjective, so a Moore–Penrose generalized inverse should be used when one is interested in a solution of (1.1). This can be a subject for future research. Consider the null space of the coefficient matrix $\mathcal{K}_\nu(z)$,

$$\operatorname{Ker} \mathcal{K}_\nu(z) = \operatorname{span} \begin{pmatrix} \mathcal{U}(z) \\ J_{(q-1)(\nu+1)} \end{pmatrix},$$

where $J_{(q-1)(\nu+1)}$ is the $(q - 1)(\nu + 1)$ rotation matrix. An algorithm of the matrix $\mathcal{U}(z)$ contained in $\operatorname{Ker}(\mathcal{K}_\nu(z))$ will be set forth to obtain

$$\mathcal{U}(z) = \frac{1}{\nu!}\left(\mathcal{U}_0(z), \mathcal{U}_1(z), \mathcal{U}_2(z), \ldots, \mathcal{U}_{\nu-1}(z)\right).$$

In this section no proofs are provided since they are similar to the proofs done in section 3.

**6.1. A representation for $\mathcal{U}_0(z)$.** An appropriate partition is $\mathcal{U}_0(z) = (\mathcal{U}_0^{(1)}(z)\ \mathcal{U}_0^{(2)}(z))$. For evaluating $\mathcal{U}_0^{(1)}(z)$ we introduce the $(\nu + 1) \times q$ matrix

$$\Omega = (\xi, \xi, \ldots, \xi), \tag{6.1}$$

where the vector $\xi$ is given in (3.4), and we put

$$\mathcal{U}_0^{(1)}(z) = \Omega \odot z^{-(q-\nu-1)}\left(u_{\nu+1}(z) u_q^\top(z)\right), \tag{6.2}$$

for $\nu = 1, 2, \ldots, q-2$. The signs of the elements of each column vector of $\mathcal{U}_0^{(1)}(z)$ follow the same pattern as for $\mathcal{U}_0(z)$ in section 3. The second part of $\mathcal{U}_0(z)$ is

$$\mathcal{U}_0^{(2)}(z) = \chi \odot \mathcal{U}_{1,2}^*(z), \tag{6.3}$$

where

$$\mathcal{U}_{1,2}^*(z) = \begin{cases} \mathcal{U}_1^*(z) & \text{for } \nu = 2, 3, \ldots, q-2 \\ \mathcal{U}_2^*(z) & \text{for } \nu = 1 \end{cases}$$

and

$$\begin{cases} \mathcal{U}_1^*(z) = u_{q-\nu-1}^{*\top}(z^{-1}) \otimes \begin{pmatrix} u_3^*(z^{-1}) \\ z u_{\nu-2}(z) \end{pmatrix} & \text{for } \nu = 2, 3, \ldots, q-2 \\ \mathcal{U}_2^*(z) = u_{q-\nu-1}^{*\top}(z^{-1}) \otimes z^{-1} u_2^*(z^{-1}) & \text{for } \nu = 1. \end{cases}$$

The matrix $\chi$ has the form $\chi = (\chi_{q-\nu-1}, \chi_{q-\nu-2}, \ldots, \chi_2, \chi_1)$, where the columns are computed recursively for $k = 2, 3, \ldots, q-\nu-1$:

$$\chi_k = \chi_{k-1} + \xi. \tag{6.4}$$

The $(\nu+1)$ column vector $\chi_1$ is for $\nu = 1, 2, \ldots, q-2$

$$\chi_1 = \begin{pmatrix} \binom{\nu}{0} \\ \binom{\nu}{1} + \binom{\nu-1}{0} \\ \binom{\nu}{2} + \binom{\nu-1}{1} \\ \vdots \\ \binom{\nu}{\nu} + \binom{\nu-1}{\nu-1} \end{pmatrix}. \tag{6.5}$$

The sign pattern of each column of $\mathcal{U}_0^{(2)}(z)$ is $(-1)^\ell$ with $\ell = 0, 1, \ldots, \nu$. In the next section we shall summarize the construction of $\mathcal{U}_0(z)$.

### 6.1.1. Summary of the construction of $\mathcal{U}_0(z)$.

*Step* 1. Introduce the vector $\xi$ according to (3.4).

*Step* 2. Define matrix $\Omega$ according to (6.1).

*Step* 3. Define the columns of $\mathcal{U}_0^{(1)}(z)$ according to (6.2).

*Step* 4. Introduce the vector $\chi_1$ given in (6.5).

*Step* 5. Compute the vectors $\chi_2, \chi_3, \ldots, \chi_{q-\nu-1}$ by means of the recursions (6.4).

*Step* 6. Compute the columns of $\mathcal{U}_0^{(2)}(z)$ according to (6.3).

### 6.2. Example for $\mathcal{U}_0(z)$ when $q = 6$, $\nu = 4$.

An example is chosen when $q = 6$ and $\nu = 4$ so the first matrices to consider are $\mathcal{U}_0^{(1)}(\sigma)$ and $\mathcal{U}_0^{(2)}(\sigma)$ to obtain

$$\mathcal{U}_0^{(1)}(z) = \begin{pmatrix} -\frac{1}{z} & -1 & -z & -z^2 & -z^3 & -z^4 \\ 4 & 4z & 4z^2 & 4z^3 & 4z^4 & 4z^5 \\ -6z & -6z^2 & -6z^3 & -6z^4 & -6z^5 & -6z^6 \\ 4z^2 & 4z^3 & 4z^4 & 4z^5 & 4z^6 & 4z^7 \\ -z^3 & -z^4 & -z^5 & -z^6 & -z^7 & -z^8 \end{pmatrix}$$

and

$$\mathcal{U}_0^{(2)}(z) = \begin{pmatrix} \frac{1}{z^2} \\ -\frac{5}{z} \\ 9 \\ -7z \\ 2z^2 \end{pmatrix}.$$

**6.3. A representation of $\mathcal{U}_j(z)$ when $j = 1, 2, \ldots, \nu - 1$.** The matrices $\mathcal{U}_1(z), \mathcal{U}_2(z), \ldots, \mathcal{U}_{\nu-1}(z)$ are now considered to obtain for $j = 1,\, 2, \ldots, \nu - 1$

$$\mathcal{U}_j(z) = \left(\mathcal{U}_j^{(1)}(z)\, \mathcal{U}_j^{(2)}(z) \mathcal{U}_j^{(3)}(z)\right).$$

Since the submatrices $\mathcal{U}_j^{(1)}(z)$ and $\mathcal{U}_j^{(2)}(z)$ have the same structure as the corresponding submatrices in section 3, the case $q = \nu + 1$, we therefore omit the description of $\mathcal{U}_j^{(1)}(z)$ and $\mathcal{U}_j^{(2)}(z)$.

**6.3.1. A representation of $\mathcal{U}_j^{(3)}(z)$.** We shall now focus on matrix $\mathcal{U}_j^{(3)}(z)$ and for that purpose the following matrix is considered for $j = 1,\, 2, \ldots,\, \nu - 1$:

$$\mu_j = \left(\mu_j^{q-\nu-1} \mu_j^{q-\nu-2} \cdots \mu_j^2 \mu_j^1\right). \tag{6.6}$$

The first recursion to consider is when $j = 1$ and $k = 2, 3, \ldots, q - \nu - 1$, to obtain

$$\mu_1^k = \mu_1^{k-1} + 2\chi_k. \tag{6.7}$$

The vectors $\chi_2, \chi_3, \ldots, \chi_{q-\nu-1}$ are obtained recursively for $\mathcal{U}_0^{(2)}(z)$; see (6.4). The solution to (6.4) is

$$\chi_k = \chi_1 + (k-1)\xi,$$

where $\chi_1$ is given in (6.5). A solution to (6.7) is then given by

$$\mu_1^k = \mu_1^1 + 2(k-1)\chi_1 + k(k-1)\xi.$$

A generalization can now be given for $j = 2, \ldots,\, \nu - 1$ and $k = 1, 2, 3, \ldots, q - \nu - 1$. The column vectors are computed recursively as follows:

$$\mu_j^k = \mu_j^{k-1} + (j+1)\, \mu_{j-1}^k. \tag{6.8}$$

A solution to recursion (6.8) in terms of initial vectors $\mu_j^1, \mu_{j-1}^1, \ldots, \mu_2^1, \mu_1^1$, specified in (6.10), and the known vectors $\chi_1$ and $\xi$, is given by

$$\begin{aligned}\mu_j^k = {} & \sum_{i=0}^{j-1} i! \binom{j+1}{i} \binom{k-2+i}{k-2} \mu_{j-i}^1 \\ & + (j+1)! \binom{k+j-2}{k-2} \chi_1 + (j+1)! \binom{k+j-1}{k-2} \xi.\end{aligned} \tag{6.9}$$

The explicit solution (6.10) is derived in a similar manner as in Proposition 3.4. For $j = 1, 2, \ldots, \nu - 1$, the components of the vector $\mu_j^1$ are given by

$$\begin{cases} \left[\mu_j^1\right]_i = (j+1)!\binom{\nu+1}{i}, & i = 0, 1, \ldots, j+1, \\ \left[\mu_j^1\right]_i = (j+1)!\binom{\nu+1}{i} - r_{i-j-2}, & i = j+2, \ldots, \nu - 1, \\ \left[\mu_j^1\right]_{\nu+1} = (j+2)!, & \end{cases} \tag{6.10}$$

where the terms $r_\ell$, are defined by

$$r_\ell = \begin{cases} (j+1)! \dbinom{\nu - j - 1}{\ell} & \text{for } \ell = 0, 1, \ldots, \nu - j - 3 \\ 0 & \text{for } \nu - j < 3. \end{cases}$$

The submatrix $\mathcal{U}_j^{(3)}(z)$ can now be given according to

$$\mathcal{U}_j^{(3)}(z) = \mu_j \odot z^{-j}\mathcal{U}_1^*(z) \qquad \text{for } \; j = 1, 2, \ldots, \nu - 1. \tag{6.11}$$

The matrix $\mathcal{U}_1^*(z)$ has also been used for specifying $\mathcal{U}_0^{(2)}(z)$. The sign pattern of the elements of each column of $\mathcal{U}_j^{(3)}(\sigma)$ follows the ordering $(-1)^{\ell+j}$ with $\ell = 0, 1, \ldots, \nu$.

**6.3.2. Summary of the construction of matrix $\boldsymbol{\mathcal{U}_j^{(3)}(\sigma)}$.** *Step* 1. Define the initial vector $\mu_j^1$ displayed in (6.10) for $j = 1, 2, \ldots, \nu - 1$.

*Step* 2. Compute the columns of matrix (6.6) by applying recursions (6.8) for the corresponding values of $j = 1, 2, \ldots, \nu - 1$.

*Step* 3. Compute the columns of matrix $\mathcal{U}_j^{(3)}(z)$ according to (6.11) for the corresponding values of $j = 1, 2, \ldots, \nu - 1$.

**6.4. Example $\boldsymbol{\mathcal{U}_j(z)}$ when $\boldsymbol{q = 6}$, $\boldsymbol{\nu = 4}$ and $\boldsymbol{j = 1, 2, 3}$.** The matrix $\mathcal{U}_j^{(1)}(z) = (\delta_j^1(z)\delta_j^2(z)\cdots\delta_j^{j+1}(z))$ will be illustrated for $j = 1,\ 2, 3$, to obtain

$$\mathcal{U}_{j=1}^{(1)}(z) = \begin{pmatrix} 0 & -1 \\ -1 & 3z \\ 3z & -3z^2 \\ -3z^2 & z^3 \\ z^3 & 0 \end{pmatrix}, \quad \mathcal{U}_{j=2}^{(1)}(z) = \begin{pmatrix} 0 & 0 & -2 \\ 0 & -2 & 4z \\ -2 & 4z & -2z^2 \\ 4z & -2z^2 & 0 \\ -2z^2 & 0 & 0 \end{pmatrix},$$

and

$$\mathcal{U}_{j=3}^{(1)}(z) = \begin{pmatrix} 0 & 0 & 0 & -6 \\ 0 & 0 & -6 & 6z \\ 0 & -6 & 6z & 0 \\ -6 & 6z & 0 & 0 \\ 6z & 0 & 0 & 0 \end{pmatrix}.$$

The matrix $\mathcal{U}_j^{(2)}(z)$ is, for $j = 1,\ 2, 3$,

$$\mathcal{U}_{j=1}^{(2)}(z) = \begin{pmatrix} -2z & -3z^2 & -4z^3 \\ 7z^2 & 11z^3 & 15z^4 \\ -9z^3 & -15z^4 & -21z^5 \\ 5z^4 & 9z^5 & 13z^6 \\ -z^5 & -2z^6 & -3z^7 \end{pmatrix}, \quad \mathcal{U}_{j=2}^{(2)}(z) = \begin{pmatrix} -6z & -12z^2 \\ 18z^2 & 40z^3 \\ -20z^3 & -50z^4 \\ 10z^4 & 28z^5 \\ -2z^5 & -6z^6 \end{pmatrix},$$

$$\mathcal{U}_{j=3}^{(2)}() = \begin{pmatrix} -24z \\ 60z^2 \\ -60z^3 \\ 30z^4 \\ -6z^5 \end{pmatrix}.$$

The matrix $\mathcal{U}_j^{(3)}(z)$ is, for $j = 1, 2, 3$,

$$\mathcal{U}_{j=1}^{(3)}(z) = \begin{pmatrix} -\frac{2}{z^3} \\ \frac{10}{z^2} \\ -\frac{20}{z} \\ 18 \\ -6z \end{pmatrix}, \quad \mathcal{U}_{j=2}^{(3)}(z) = \begin{pmatrix} \frac{6}{z^4} \\ -\frac{30}{z^3} \\ \frac{60}{z^2} \\ -\frac{60}{z} \\ 24 \end{pmatrix},$$

and

$$\mathcal{U}_{j=3}^{(3)}(z) = \begin{pmatrix} -\frac{24}{z^5} \\ \frac{120}{z^4} \\ -\frac{240}{z^3} \\ \frac{240}{z^2} \\ -\frac{120}{z} \end{pmatrix}.$$

Insertion in (3.2) of the matrices $\mathcal{U}_0^{(1)}(z)$ and $\mathcal{U}_0^{(2)}(z)$, followed by the matrices

$$\begin{aligned} \mathcal{U}_1(z) &= \left( \mathcal{U}_{j=1}^{(1)}(z)\, \mathcal{U}_{j=1}^{(2)}(z)\, \mathcal{U}_{j=1}^{(3)}(z) \right), \\ \mathcal{U}_2(z) &= \left( \mathcal{U}_{j=2}^{(1)}(z)\, \mathcal{U}_{j=2}^{(2)}(z)\, \mathcal{U}_{j=2}^{(3)}(z) \right), \\ \mathcal{U}_3(z) &= \left( \mathcal{U}_{j=3}^{(1)}(z)\, \mathcal{U}_{j=3}^{(2)}(z)\, \mathcal{U}_{j=3}^{(3)}(z) \right), \end{aligned}$$

results in the scheme

$$\mathcal{U}(z) = \frac{1}{4!} \left( \mathcal{U}_0^{(1)}(z), \mathcal{U}_0^{(2)}(z), \mathcal{U}_1(z), \mathcal{U}_2(z), \mathcal{U}_3(z) \right).$$

The columns that compose the matrix $\binom{\mathcal{U}(z)}{J_{25}}$ span the null space of $\mathcal{K}_\nu(z)$ when $q = 6$ and $\nu = 4$.

## 7. Conclusions.

In this paper a solution to new linear systems of equations is displayed. This is done when $q = \nu + 1$ and $p = \tau + 1$. The newly developed algorithms for the null space and right-inverse are then equivalent for both coefficient matrices. Explicit solutions to both linear system of equations can then be straightforwardly implemented by using the same algorithms. The algorithms for the null space do not require matrix multiplications and matrix inversions. The main computational exercise consists of evaluating factorials and binomial coefficients combined with recursions that consist of the addition of two vectors. The binomial coefficients can be computed by applying the Pascal triangle.

A connection between adjoints of companion-related matrices and rectangular generalized Vandermonde matrices of the block Toeplitz type is then confirmed through the corresponding null spaces.

An algorithm for the null space for $\mathcal{K}_\nu(z)$ is also set forth when $q > \nu + 1$. To compute a solution to the linear systems of (1.1) and (1.2) under these conditions can be considered for future research.

**Acknowledgments.** The authors would like to thank the associate editor Dario Bini and one of the referees for their comments that have led to several important improvements of this paper. The authors also thank Guy Mélard for valuable suggestions.

## REFERENCES

[1] I. Gohberg, T. Kailath, and I. Koltracht, *Efficient solution of linear systems of equations with recursive structure*, Linear Algebra Appl., 80 (1986), pp. 81–113.

[2] I. Gohberg, T. Kailath, I. Koltracht, and P. Lancaster, *Linear complexity parallel algorithms for linear systems of equations with recursive structure*, Linear Algebra Appl., 88 (1987), pp. 271–315.

[3] U. Grenander and G. Szegő, *Toeplitz Forms and their Applications*, 2nd ed., Chelsea Publishing Co., New York, 1984.

[4] G. Heinig and K. Rost, *Recursive solution of Cauchy-Vandermonde systems of equations*, Linear Algebra Appl., 218 (1995), pp. 59–72.

[5] A. Klein and P. Spreij, *On the solution of Stein's equation and Fisher's information matrix of an ARMAX process*, Linear Algebra Appl., 396 (2005), pp. 1–34.

[6] A. Klein and P. Spreij, *Some results on Vandermonde matrices with an application to time series analysis*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 213–223.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1218–1241

# BACKWARD ERROR OF POLYNOMIAL EIGENPROBLEMS SOLVED BY LINEARIZATION*

NICHOLAS J. HIGHAM†, REN-CANG LI‡, AND FRANÇOISE TISSEUR†

**Abstract.** The most widely used approach for solving the polynomial eigenvalue problem $P(\lambda)x = \bigl(\sum_{i=0}^{m} \lambda^i A_i\bigr)x = 0$ in $n \times n$ matrices $A_i$ is to linearize to produce a larger order pencil $L(\lambda) = \lambda X + Y$, whose eigensystem is then found by any method for generalized eigenproblems. For a given polynomial $P$, infinitely many linearizations $L$ exist and approximate eigenpairs of $P$ computed via linearization can have widely varying backward errors. We show that if a certain one-sided factorization relating $L$ to $P$ can be found then a simple formula permits recovery of right eigenvectors of $P$ from those of $L$, and the backward error of an approximate eigenpair of $P$ can be bounded in terms of the backward error for the corresponding approximate eigenpair of $L$. A similar factorization has the same implications for left eigenvectors. We use this technique to derive backward error bounds depending only on the norms of the $A_i$ for the companion pencils and for the vector space $\mathbb{DL}(P)$ of pencils recently identified by Mackey, Mackey, Mehl, and Mehrmann. In all cases, sufficient conditions are identified for an optimal backward error for $P$. These results are shown to be entirely consistent with those of Higham, Mackey, and Tisseur on the conditioning of linearizations of $P$. Other contributions of this work are a block scaling of the companion pencils that yields improved backward error bounds; a demonstration that the bounds are applicable to certain structured linearizations of structured polynomials; and backward error bounds specialized to the quadratic case, including analysis of the benefits of a scaling recently proposed by Fan, Lin, and Van Dooren. The results herein make no assumptions on the stability of the method applied to $L$ or whether the method is direct or iterative.

**Key words.** backward error, scaling, eigenvector, matrix polynomial, matrix pencil, linearization, companion form, quadratic eigenvalue problem, alternating, palindromic

**AMS subject classifications.** 65F15, 15A18

**DOI.** 10.1137/060663738

## 1. Introduction.

The polynomial eigenvalue problem (PEP) is to find scalars $\lambda$ and nonzero vectors $x$ and $y$ satisfying $P(\lambda)x = 0$ and $y^*P(\lambda) = 0$, where

$$P(\lambda) = \sum_{i=0}^{m} \lambda^i A_i, \qquad A_i \in \mathbb{C}^{n\times n}, \quad A_m \neq 0 \tag{1.1}$$

is a matrix polynomial of degree $m$. Here, $x$ and $y$ are right and left eigenvectors corresponding to the eigenvalue $\lambda$. We will assume throughout that $P$ is regular, that is, $\det P(\lambda) \not\equiv 0$.

The standard way of solving this problem is to convert $P$ into a linear polynomial

$$L(\lambda) = \lambda X + Y, \qquad X, Y \in \mathbb{C}^{mn\times mn}$$

---

*Received by the editors June 26, 2006; accepted for publication (in revised form) by P. Van Dooren June 7, 2007; published electronically December 7, 2007.
http://www.siam.org/journals/simax/29-4/66373.html

†School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/, ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/). The work of the first author was supported by a Royal Society-Wolfson Research Merit Award. The work of the first and third authors was supported by Engineering and Physical Sciences Research Council grants GR/S31693 and EP/D079403.

‡Department of Mathematics, University of Texas at Arlington, P.O. Box 19408, Arlington, TX 76019 (rcli@uta.edu). The work of this author was supported in part by the National Science Foundation CAREER award under grant CCR-9875201 and by the National Science Foundation under grant DMS-0510664.

with the same spectrum as $P$ and solve the eigenproblem for $L$. This generalized eigenproblem is usually solved with the QZ algorithm [20] for small to medium size problems or a projection method for large sparse problems [1]. That $L$ has the same spectrum as $P$ is assured if

$$E(\lambda)L(\lambda)F(\lambda) = \begin{bmatrix} P(\lambda) & 0 \\ 0 & I_{(m-1)n} \end{bmatrix} \tag{1.2}$$

for some unimodular $E(\lambda)$ and $F(\lambda)$. (A matrix polynomial $E(\lambda)$ is unimodular if its determinant is a nonzero constant, independent of $\lambda$.) Such an $L$ is called a *linearization* of $P(\lambda)$ [5, sec. 7.2]. As an example, the pencil

$$C_1(\lambda) = \lambda \begin{bmatrix} A_3 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} + \begin{bmatrix} A_2 & A_1 & A_0 \\ -I & 0 & 0 \\ 0 & -I & 0 \end{bmatrix} \tag{1.3}$$

can be shown to be a linearization for the cubic $P(\lambda) = \lambda^3 A_3 + \lambda^2 A_2 + \lambda A_1 + A_0$; it is known as the *first companion* linearization.

Among the infinitely many linearizations $L$ of $P$ we are interested in those whose right and left eigenvectors permit easy recovery of the corresponding eigenvectors of $P$. For example, if $(x, y)$ and $(z, w)$ denote pairs of right and left eigenvectors of the cubic $P(\lambda)$ and its companion linearization $C_1(\lambda)$, respectively, associated with the simple, finite eigenvalue $\lambda$, then

$$(z, w) = \left( \begin{bmatrix} \lambda^2 x \\ \lambda x \\ x \end{bmatrix}, \begin{bmatrix} y \\ (\bar{\lambda} A_3^* + A_2^*)y \\ (\bar{\lambda}^2 A_3^* + \bar{\lambda} A_2^* + A_1^*)y \end{bmatrix} \right), \tag{1.4}$$

so that $x$ can be recovered from one of the first two blocks (if $\lambda \neq 0$) or the third block of $n$ components of $z$, and $y$ can be recovered from the first $n$ components of $w$. This correspondence extends to all eigenvalues and arbitrary $m$, as we will explain in section 3.

In practice, the eigenpairs of $L$ are not computed exactly because of rounding errors and, in the case of iterative methods, truncation errors. For a given approximate eigenpair of $L$, it is important to know how good an approximate eigenpair of $P$ will be produced. Here, "good" can have various meanings; in particular, it can refer to the relative error of the eigenvalue or the backward error of the eigenpair. The relative error question has been investigated by Higham, Mackey, and Tisseur [7], by analyzing the conditioning of both the polynomial $P$ and the linearization $L$. The purpose of the present work is to investigate the backward error for a wide variety of linearizations. Two key aspects of this task can be seen by considering the companion pencil (1.3). First, a small but arbitrary perturbation to $C_1$, such as that introduced by the QZ algorithm, does not respect the zero and identity blocks and so may not correspond to a small perturbation of $P$. Second, the block from which the approximate eigenvector is recovered will influence the backward error.

Our work builds on that of Tisseur [22], who shows that solving a quadratic eigenvalue problem (QEP) by applying a numerically stable method to the companion linearization can be backward unstable, but that stability is guaranteed if all the coefficient matrices have unit norm.

In section 2.1 we define the backward error $\eta_P$ of an approximate eigenpair and eigentriple of $P$ for the polynomial both in the $\lambda$-form (1.1) and in homogeneous $(\alpha, \beta)$-form. In section 2.2 we show that given appropriate one-sided factorizations relating a linearization $L$ to the original polynomial $P$, we can bound the backward error of an approximate eigenpair of $P$ in terms of the backward error of the approximate eigenpair of $L$ from which it was obtained. The bounds have the useful feature of separating the dependence on $L$, $P$, and $(\alpha, \beta)$ from the dependence on how the (right or left) eigenvector is recovered.

In section 3 we introduce the first and second companion linearizations and the vector spaces $\mathbb{L}_1$ and $\mathbb{L}_2$ of pencils associated with $P$. As a by-product of our analysis we obtain in section 3.1 new formulae for recovering a left (right) eigenvector of $P$ from one of a linearization in $\mathbb{L}_1$ ($\mathbb{L}_2$). In section 3.2 we obtain backward error bounds for the companion pencils and deduce sufficient conditions for a small backward error $\eta_P$. We show in section 3.3 that applying a block scaling to the companion pencils yields smaller backward error bounds when $\max_i \|A_i\|_2$ is much different from 1. The vector space $\mathbb{DL}(P) = \mathbb{L}_1(P) \cap \mathbb{L}_2(P)$ is then considered in section 3.4, where bounds of the same form as for the block-scaled companion pencils are obtained. In section 3.5 we explain how the backward error results provide essentially the same guidance on optimal choice of linearizations as the condition number bounds of Higham, Mackey, and Tisseur [7]. In section 4 we show that the results of section 3 also apply to certain structured linearizations of structured polynomials.

The special case of quadratic polynomials $\lambda^2 A + \lambda B + C$ is studied in detail in section 5, concentrating on the companion linearization and the $\mathbb{DL}(P)$ basis pencils $L_1$ and $L_m$. Bounds for $\eta_P$ are obtained and then specialized to exploit a scaling procedure recently proposed by Fan, Lin, and Van Dooren [2]. The bounds involve a growth factor $\omega$ that is shown to be bounded by $1+\tau$, where $\tau = \|B\|_2/\sqrt{\|A\|_2\|C\|_2}$. Our analysis improves upon that in [2], which contains a growth term $\max(1+\tau, 1+\tau^{-1})$. The bounds are particularly satisfactory for elliptic QEPs and, more generally, QEPs that are not too heavily damped. Numerical experiments illustrating these and other aspects of the theory are given in section 6.

Finally, we note that our results are of interest even in the case $n = 1$, although we will not consider this case specifically here. The roots of a scalar polynomial, $p$, are often found by computing the eigenvalues of a corresponding companion matrix, $C$. Our analysis provides new bounds on the backward errors of the computed roots of $p$ in terms of the backward errors of the computed eigenvalues of $C$.

## 2. Backward errors.

**2.1. Definition and notation.** The normwise backward error of an approximate (right) eigenpair $(x, \lambda)$ of $P(\lambda)$, where $\lambda$ is finite, is defined by

$$\eta_P(x,\lambda) = \min\{\, \epsilon : (P(\lambda) + \Delta P(\lambda))x = 0,\ \|\Delta A_i\|_2 \le \epsilon \|A_i\|_2,\ i = 0{:}\,m \,\}, \tag{2.1}$$

where $\Delta P(\lambda) = \sum_{i=0}^{m} \lambda^i \Delta A_i$. Tisseur [22, Thm. 1] obtained the explicit formula

$$\eta_P(x,\lambda) = \frac{\|P(\lambda)x\|_2}{\left(\sum_{i=0}^{m} |\lambda^i| \|A_i\|_2\right) \|x\|_2}. \tag{2.2}$$

Similarly, for an approximate left eigenpair $(y^*, \lambda)$, we have

$$\eta_P(y^*,\lambda) := \min\{\, \epsilon : y^*(P(\lambda) + \Delta P(\lambda)) = 0,\ \|\Delta A_i\|_2 \le \epsilon \|A_i\|_2,\ i = 0{:}\,m \,\} \tag{2.3}$$

$$= \frac{\|y^* P(\lambda)\|_2}{\left(\sum_{i=0}^{m} |\lambda^i| \|A_i\|_2\right) \|y\|_2}. \tag{2.4}$$

Also of interest is the backward error of the approximate triplet $(x, y, \lambda)$ [22, Thm. 4]:

$$\begin{aligned} (2.5)\quad \eta_P(x, y^*, \lambda) &:= \min\{\, \epsilon : (P(\lambda) + \Delta P(\lambda))x = 0,\ y^*(P(\lambda) + \Delta P(\lambda)) = 0, \\ &\qquad \|\Delta A_i\|_2 \le \epsilon \|A_i\|_2,\ i = 0\colon m \,\} \\ (2.6)\qquad\qquad &= \max\bigl(\eta_P(x, \lambda),\ \eta_P(y^*, \lambda)\bigr). \end{aligned}$$

We make two comments on notation. As an argument of $\eta$, a left eigenvector is written as a row vector to distinguish it from a right eigenvector. Symbols such as $\lambda$, $x$, and $y$ will denote both exact and (more often) approximate quantities, with the context making clear which usage is in effect. The alternative of using a tilde to denote approximate quantities leads to rather cumbersome formulae.

In order to define backward errors valid for all $\lambda$, including $\infty$, we rewrite the polynomial in the homogeneous form

$$P(\alpha, \beta) = \sum_{i=0}^{m} \alpha^i \beta^{m-i} A_i$$

and identify $\lambda$ with any pair $(\alpha, \beta) \ne (0, 0)$ for which $\lambda = \alpha/\beta$. The definitions (2.1), (2.3), and (2.5) are trivially rewritten in terms of $\alpha$ and $\beta$. Using $P(\alpha, \beta) = \beta^m P(\alpha/\beta)$ for $\beta \ne 0$, we find that in place of (2.2), (2.4), and (2.6) we have

$$\eta_P(x, \alpha, \beta) = \frac{\|P(\alpha, \beta)x\|_2}{\left(\sum_{i=0}^{m} |\alpha|^i |\beta|^{m-i} \|A_i\|_2\right) \|x\|_2}, \tag{2.7}$$

$$\eta_P(y^*, \alpha, \beta) = \frac{\|y^* P(\alpha, \beta)\|_2}{\left(\sum_{i=0}^{m} |\alpha|^i |\beta|^{m-i} \|A_i\|_2\right) \|y\|_2}, \tag{2.8}$$

$$\eta_P(x, y^*, \alpha, \beta) = \max\bigl(\eta_P(x, \alpha, \beta),\ \eta_P(y^*, \alpha, \beta)\bigr). \tag{2.9}$$

Note that these expressions are independent of the choice of $\alpha$ and $\beta$ representing the eigenvalue; that is, a scaling $\alpha \leftarrow \theta\alpha$, $\beta \leftarrow \theta\beta$ with $\theta \ne 0$ leaves the expressions unchanged.

### 2.2. Bounding the backward error for $P$ relative to that for $L$.

Let $L(\lambda) = \lambda X + Y$ be a linearization of $P(\lambda)$. For approximate right eigenvectors $z$ of $L$ and $x$ of $P$, both corresponding to an approximate eigenvalue$(\alpha, \beta)$, our aim is to compare $\eta_P(x, \alpha, \beta)$ with

$$\eta_L(z, \alpha, \beta) = \frac{\|L(\alpha, \beta)z\|_2}{(|\alpha| \|X\|_2 + |\beta| \|Y\|_2)\|z\|_2}, \tag{2.10}$$

which is obtained by applying (2.7) to $L(\alpha, \beta) = \alpha X + \beta Y$. Of course, this comparison is possible only if there is some well-defined relation between $x$ and $z$. Such a relation, and a means for bounding $\eta_P$, both follow from one key assumption: that we can find an $n \times nm$ matrix polynomial $G(\alpha, \beta)$ such that

$$G(\alpha, \beta)L(\alpha, \beta) = g^T \otimes P(\alpha, \beta) \tag{2.11}$$

for some nonzero $g \in \mathbb{C}^m$, where $\otimes$ denotes the Kronecker product [15, sec. 12.1]. Necessarily, $G(\alpha, \beta)$ will have degree $m - 1$. Note that this is a one-sided transformation as opposed to the two-sided transformation in the definition of linearization. Then we have

$$G(\alpha, \beta)L(\alpha, \beta)z = \bigl(g^T \otimes P(\alpha, \beta)\bigr)z = P(\alpha, \beta)(g^T \otimes I_n)z, \tag{2.12}$$

where the latter equation relies on $g^T$ being a row vector. Thus if $z$ is an eigenvector of $L$ then

$$x := (g^T \otimes I_n)z = \sum_{i=1}^{m} g_i z_i, \qquad z_i := z((i-1)n+1\colon in) \tag{2.13}$$

is an eigenvector of $P$, provided that $x$ is nonzero. This latter requirement is not satisfied in general but will be proved for some important classes of linearizations. As an example, for the first companion linearization $C_1(\alpha,\beta) = \beta^3 C_1(\alpha/\beta)$ in (1.3), it is easily checked that $G(\alpha,\beta) = [\,\alpha^2 I \quad -(\beta^2 A_0 + \alpha\beta A_1) \quad -\alpha\beta A_0\,]$ satisfies (2.11) with $g = e_1$, the first column of the identity matrix, and that if $z$ is a right eigenvector of $C_1$ and $\alpha \neq 0$ then $x = z_1 = z(1\colon n) \neq 0$ is a right eigenvector for $P$ (cf. (1.4)).

Suppose now that (2.11) is satisfied, an approximate right eigenvector $z$ of $L$ is given, and $x$ is given by (2.13). Then, by (2.7), (2.10), and (2.12),

$$\begin{aligned} \eta_P(x,\alpha,\beta) &\leq \frac{\|G(\alpha,\beta)\|_2 \|L(\alpha,\beta)z\|_2}{\left(\sum_{i=0}^{m} |\alpha|^i |\beta|^{m-i} \|A_i\|_2\right) \|x\|_2} \\ &\leq \frac{|\alpha|\,\|X\|_2 + |\beta|\,\|Y\|_2}{\sum_{i=0}^{m} |\alpha|^i |\beta|^{m-i} \|A_i\|_2} \cdot \frac{\|G(\alpha,\beta)\|_2 \|z\|_2}{\|x\|_2} \cdot \eta_L(z,\alpha,\beta). \end{aligned} \tag{2.14}$$

This bound largely separates the dependence on $L$, $P$, and $(\alpha,\beta)$ (in the first term) from the dependence on $G$ and $z$ (in the second term).

For left eigenvectors the appropriate analogue of the assumption (2.11) is that there exists an $mn \times n$ matrix polynomial $H(\alpha,\beta)$ such that

$$L(\alpha,\beta)H(\alpha,\beta) = h \otimes P(\alpha,\beta) \tag{2.15}$$

for some nonzero $h \in \mathbb{C}^m$. We then have, for $w \in \mathbb{C}^{mn}$,

$$w^* L(\alpha,\beta)H(\alpha,\beta) = w^*(h \otimes P(\alpha,\beta)) = w^*(h \otimes I_n)P(\alpha,\beta). \tag{2.16}$$

Hence if $w$ is a left eigenvector of $L$ then

$$y := (h^* \otimes I_n)w = \sum_{i=1}^{m} \overline{h}_i w_i, \qquad w_i := w((i-1)n+1\colon in) \tag{2.17}$$

is a left eigenvector of $P$, provided that it is nonzero. From (2.8) and (2.17) we obtain for an approximate left eigenvector $w$ of $L$ the bound

$$\eta_P(y^*,\alpha,\beta) \leq \frac{|\alpha|\|X\|_2 + |\beta|\|Y\|_2}{\sum_{i=0}^{m} |\alpha|^i |\beta|^{m-i} \|A_i\|_2} \cdot \frac{\|H(\alpha,\beta)\|_2 \|w\|_2}{\|y\|_2} \cdot \eta_L(w^*,\alpha,\beta). \tag{2.18}$$

In the rest of this paper we show that one or both of assumptions (2.11) and (2.15) are satisfied for a wide class of linearizations, and we study the upper bounds (2.14) and (2.18).

## 3. Unstructured linearizations.

We first concentrate on general, unstructured matrix polynomials, treating companion and $\mathbb{DL}(P)$ linearizations.

Associated with $P$ are two companion pencils, $C_1(\lambda) = \lambda X_1 + Y_1$ and $C_2(\lambda) = \lambda X_2 + Y_2$, called the first and second companion forms [15, sec. 14.1], respectively,

where

$$X_1 = X_2 = \operatorname{diag}(A_m, I_n, \dots, I_n),$$

$$Y_1 = \begin{bmatrix} A_{m-1} & A_{m-2} & \dots & A_0 \\ -I_n & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & -I_n & 0 \end{bmatrix}, \quad Y_2 = \begin{bmatrix} A_{m-1} & -I_n & \dots & 0 \\ A_{m-2} & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -I_n \\ A_0 & 0 & \dots & 0 \end{bmatrix}. \tag{3.1}$$

They are widely used in practice. For example, the MATLAB function `polyeig` that solves the PEP uses the reversed first companion linearization $\operatorname{rev}C_1(\lambda)$ of the reversed matrix polynomial $\operatorname{rev}P(\lambda)$. The reversal operator is defined for $P$ in (1.1) by

$$\operatorname{rev}P(\lambda) = \lambda^m P(1/\lambda) = \sum_{i=0}^{m} \lambda^i A_{m-i}. \tag{3.2}$$

The companion forms have the important property that they are always linearizations [19, sec. 4].

$C_1(\lambda)$ and $C_2(\lambda)$ belong to large sets of potential linearizations recently identified by Mackey et al. [19] and studied in [6] and [19]. With the notation $\Lambda = [\lambda^{m-1}, \lambda^{m-2}, \dots, 1]^T$, these sets are

$$\mathbb{L}_1(P) = \{\, L(\lambda) : L(\lambda)(\Lambda \otimes I_n) = v \otimes P(\lambda),\ v \in \mathbb{C}^m \,\}, \tag{3.3}$$

$$\mathbb{L}_2(P) = \{\, L(\lambda) : (\Lambda^T \otimes I_n)L(\lambda) = \widetilde{v}^T \otimes P(\lambda),\ \widetilde{v} \in \mathbb{C}^m \,\}. \tag{3.4}$$

There are many $L(\lambda) \in \mathbb{L}_1(P)$ corresponding to a given $v$, and likewise for $\mathbb{L}_2(P)$; indeed, $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$ both have dimension $m(m-1)n^2 + m$ [19, Cor. 3.6]. It is easy to check that $C_1(\lambda)$ and $C_2(\lambda)$ belong to $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$, respectively, with $\widetilde{v} = v = e_1$; so the pencils in $\mathbb{L}_1$ and $\mathbb{L}_2$ can be thought of as generalizations of the first and second companion forms. It is proved in [19, Prop. 3.2, Prop. 3.12, Thm. 4.7] that $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$ are vector spaces and that almost all pencils in these spaces are linearizations of $P$.

One of the underlying reasons for the interest in $\mathbb{L}_1$ and $\mathbb{L}_2$ is that eigenvectors of $P$ can be directly recovered from eigenvectors of linearizations in $\mathbb{L}_1$ and $\mathbb{L}_2$. As with the backward errors, it is more convenient to use the $(\alpha, \beta)$ notation, so we define

$$\Lambda_{\alpha,\beta} = [\alpha^{m-1}, \alpha^{m-2}\beta, \dots, \beta^{m-1}]^T = \beta^{m-1}\Lambda.$$

THEOREM 3.1 (eigenvector recovery from $\mathbb{L}_1$ and $\mathbb{L}_2$).

- *If $L \in \mathbb{L}_1(P)$ is a linearization of $P$ then every right eigenvector of $L$ with eigenvalue $(\alpha, \beta)$ is of the form $\Lambda_{\alpha,\beta} \otimes x$ for some right eigenvector $x$ of $P$.*
- *If $L \in \mathbb{L}_2(P)$ is a linearization of $P$ then every left eigenvector of $L$ with eigenvalue $(\alpha, \beta)$ is of the form $\overline{\Lambda}_{\alpha,\beta} \otimes y$ for some left eigenvector $y$ of $P$.*

*Proof.* See [19, Thms. 3.8, 3.14, 4.4]. ☐

Theorem 3.1 shows that from any right eigenvector $z$ of $L \in \mathbb{L}_1$ we can read off a right eigenvector of $P$ by looking at any nonzero subvector $z_i = z((i-1)n+1\colon in)$, and similarly a left eigenvector of $L \in \mathbb{L}_2$ yields a left eigenvector of $P$.

### 3.1. $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$.

It is immediate from (3.3) and (3.4) that the pencils in $\mathbb{L}_1$ satisfy (2.15), while those in $\mathbb{L}_2$ satisfy (2.11). Therefore our backward error bounds are applicable to left eigenvectors of pencils in $\mathbb{L}_1$ and right eigenvectors of pencils in $\mathbb{L}_2$, provided that the vectors $x$ in (2.13) and $y$ in (2.17) are nonzero when $z$ and $w$ are exact eigenvectors. In fact, for $\mathbb{L}_1$ and $\mathbb{L}_2$ (2.13) and (2.17) define a bijection between eigenvectors of the pencil and of $P$ and so allow recovery of *all* the eigenvectors. The following two new results supplement the existing eigenvector recovery formulae in Theorem 3.1.

THEOREM 3.2 (left eigenvector recovery from $\mathbb{L}_1$). *Let $L \in \mathbb{L}_1(P)$ be a linearization of $P$, with vector $v$ (necessarily nonzero) in* (3.3). *If $w$ is a left eigenvector of $L$ with eigenvalue $(\alpha, \beta)$ then*

$$y = (v^* \otimes I_n)w \tag{3.5}$$

*is a left eigenvector of $P$ with eigenvalue $(\alpha, \beta)$. Moreover, any left eigenvector of $P$ corresponding to $(\alpha, \beta)$ can be recovered from one of $L$ from the formula* (3.5).

*Proof.* Assume, first, that $\mu \equiv (\alpha, \beta)$ is finite. For arbitrary $\lambda$, premultiplying the condition defining $\mathbb{L}_1$ by $w^*$ (or simply using the $\lambda$-analogue of (2.16)) gives

$$w^* L(\lambda)(\Lambda \otimes I_n) = w^*\bigl(v \otimes P(\lambda)\bigr) = w^*(v \otimes I_n)P(\lambda) =: y^* P(\lambda).$$

Since $w^* L(\mu) = 0$, it follows that $y^* P(\mu) = 0$. We therefore just have to show that $y \neq 0$. We suppose that $y = 0$ and will obtain a contradiction. If $y = 0$ then $w^* L(\lambda)(\Lambda \otimes I_n) \equiv 0$. Since $L$ is linear, we can write

$$w^* L(\lambda) = [b_1(\lambda),\ b_2(\lambda), \dots, b_m(\lambda)],$$

where $b_i(\lambda) = c_i \lambda + d_i \in \mathbb{C}^{1 \times n}$ is linear. Then

$$\begin{aligned} 0 \equiv w^* L(\lambda)(\Lambda \otimes I_n) &= [b_1(\lambda),\ b_2(\lambda), \dots,\ b_m(\lambda)] \begin{bmatrix} \lambda^{m-1} I_n \\ \lambda^{m-2} I_n \\ \vdots \\ I_n \end{bmatrix} \\ &= \lambda^{m-1} b_1(\lambda) + \lambda^{m-2} b_2(\lambda) + \cdots + b_m(\lambda) \\ &= \lambda^m c_1 + \lambda^{m-1}(d_1 + c_2) + \cdots + \lambda(d_{m-1} + c_m) + d_m. \end{aligned}$$

Hence $c_1 = 0, d_1 = -c_2, \dots, d_{m-1} = -c_m,\ d_m = 0$. Then

$$0 = w^* L(\mu) = [b_1(\mu), b_2(\mu), \dots, b_m(\mu)] = [-c_2,\ \mu c_2 - c_3, \dots, \mu c_{m-1} - c_m,\ \mu c_m],$$

which implies $c_2 = c_3 = \cdots = c_m = 0$. Hence $b_i(\lambda) \equiv 0$ for all $i$. Thus $w^* L(\lambda) \equiv 0$, which means that $L$ is a nonregular polynomial. But by [19, Thm. 4.3], $L \in \mathbb{L}_1(P)$ being nonregular implies that $L$ is not a linearization of $P$. This is a contradiction, and so $y \neq 0$, as required.

The case $\mu = \infty$ can be handled by expressing $L$ and $P$ in homogeneous $(\alpha, \beta)$-form and using $\mu \equiv (1, 0)$. The details are a minor variation on those above.

Finally, consider the map $w \mapsto (v^* \otimes I_n)w$ from $\mathcal{K}_1 = \operatorname{left\,ker} L(\alpha, \beta)$ to $\mathcal{K}_2 = \operatorname{left\,ker} P(\alpha, \beta)$, where left ker denotes the left kernel. The first part showed that this map has kernel $\{0\}$. Since $L \in \mathbb{L}_1(P)$ is a linearization and $P$ is regular, $L$ is a strong linearization[1] [19, Thm. 4.3]. Hence the geometric multiplicity of any eigenvalue (including $\infty$) is the same for $L$ and $P$ [14]; that is, $\mathcal{K}_1$ and $\mathcal{K}_2$ have the same dimension. It follows that the map is a bijection, and the result is proved. □

---

[1] $L$ is a strong linearization of $P$ if it is a linearization for $P$ and $\operatorname{rev} L$ is a linearization for $\operatorname{rev} P$.

THEOREM 3.3 (right eigenvector recovery from $\mathbb{L}_2$). *Let* $L \in \mathbb{L}_2(P)$ *be a linearization, with vector* $\widetilde{v}$ (*necessarily nonzero*) *in* (3.4). *If* $z$ *is a right eigenvector of* $L$ *with eigenvalue* $(\alpha, \beta)$ *then*

$$x = (\widetilde{v}^T \otimes I_n)z \tag{3.6}$$

*is a right eigenvector of* $P$ *with eigenvalue* $(\alpha, \beta)$. *Moreover, any right eigenvector of* $P$ *corresponding to* $(\alpha, \beta)$ *can be recovered from one of* $L$ *from the formula* (3.6).

*Proof.* The proof is entirely analogous to that of Theorem 3.2. ◻

The broader significance of Theorems 3.2 and 3.3 combined with Theorem 3.1 is that both left *and* right eigenvectors of pencils in $\mathbb{L}_1$ and $\mathbb{L}_2$ yield corresponding eigenvectors of $P$ via simple formulae.

We will not write down backward error bounds for $\mathbb{L}_1$ and $\mathbb{L}_2$, but will do so for their intersection in section 3.4.

**3.2. Companion linearizations.** It is easy to see that $C_2(P) = C_1(P^T)^T$, where $P^T$ denotes the polynomial obtained by transposing each coefficient matrix $A_i$. This property implies that any backward error results for $C_1$ have a counterpart for $C_2$, and so it suffices to concentrate on the first companion form.

Is the factorization (2.11) possible for the first companion linearization? For $C_1$ in (1.3) with $m = 3$ it is straightforward to verify that $E(\alpha, \beta)C_1(\alpha, \beta) = I_3 \otimes P(\alpha, \beta)$ with

$$E(\alpha, \beta) = \begin{bmatrix} \alpha^2 I_n & -(\beta^2 A_0 + \alpha\beta A_1) & -\alpha\beta A_0 \\ \alpha\beta I_n & \alpha\beta A_2 + \alpha^2 A_3 & -\beta^2 A_0 \\ \beta^2 I_n & \beta^2 A_2 + \alpha\beta A_3 & \beta^2 A_1 + \alpha\beta A_2 + \alpha^2 A_3 \end{bmatrix}$$

(indeed the first block row of this equation was mentioned in section 2.2), so that we have three choices for $G(\alpha, \beta)$, namely, $G_k(\alpha, \beta) := (e_k^T \otimes I_n)E(\alpha, \beta)$, $k = 1{:}3$. This result generalizes to arbitrary degrees $m$.

LEMMA 3.4. *For the first companion form* $C_1(\alpha, \beta) = \alpha X_1 + \beta Y_1$, *for any* $m$, *there exists a block* $m \times m$ *matrix* $E(\alpha, \beta) \in \mathbb{C}^{mn \times mn}$ *such that*

$$E(\alpha, \beta)C_1(\alpha, \beta) = I_m \otimes P(\alpha, \beta), \tag{3.7}$$

*where the blocks are given by*

$$[E(\alpha, \beta)]_{i1} = \alpha^{m-i}\beta^{i-1} I_n, \qquad [E(\alpha, \beta)]_{ij} = \sum_{k=0}^{m-1} s_k \alpha^k \beta^{m-k-1} A_{\ell_k} \text{ for } j > 1,$$

*where* $s_k \in \{-1, 0, 1\}$ *and the indices* $\ell_k$ *are distinct* (*our notation suppresses the dependence of* $s_k$ *and* $\ell_k$ *on* $i$ *and* $j$). *The condition* (2.11) *is satisfied for*

$$G_k(\alpha, \beta) = (e_k^T \otimes I_n)E(\alpha, \beta), \quad g = e_k, \qquad k = 1{:}m. \tag{3.8}$$

*Proof.* The proof consists of a direct verification that $E(\alpha, \beta)$ defined by

$$[E(\alpha, \beta)]_{ij} = \begin{cases} \alpha^{m-i}\beta^{i-1} I_n, & 1 \le i \le m,\ j = 1, \\ -(\alpha/\beta)^{j-i} \displaystyle\sum_{k=0}^{m-j} \alpha^{k-1}\beta^{m-k} A_k, & 1 \le i < j,\ 1 < j \le m, \\ (\alpha/\beta)^{j-i} \displaystyle\sum_{k=m-j+1}^{m} \alpha^{k-1}\beta^{m-k} A_k, & 1 < j \le i \le m, \end{cases}$$

satisfies (3.7). ◻

The next lemma will be useful when taking norms of block matrices.

LEMMA 3.5. *For any block $\ell \times m$ matrix $B$ we have $\|B\|_2 \le \sqrt{\ell m}\,\max_{i,j} \|B_{ij}\|_2$.*

*Proof.* Partitioning $x$ conformably with $B$, we have

$$\begin{aligned}\|Bx\|_2^2 &= \sum_{i=1}^{\ell} \Bigl\| \sum_{j=1}^{m} B_{ij}x_j \Bigr\|_2^2 \le \max_{i,j} \|B_{ij}\|_2^2 \sum_{i=1}^{\ell} \Bigl( \sum_{j=1}^{m} \|x_j\|_2 \Bigr)^2 \\ &\le \max_{i,j} \|B_{ij}\|_2^2 \sum_{i=1}^{\ell} m \Bigl( \sum_{j=1}^{m} \|x_j\|_2^2 \Bigr) = \ell m \max_{i,j} \|B_{ij}\|_2^2 \, \|x\|_2^2.\end{aligned}$$

The result follows. ◻

To investigate the size of the upper bound in (2.14) for $L(\alpha,\beta) = C_1(\alpha,\beta) = \alpha X_1 + \beta Y_1$ we need to bound $\|X_1\|_2$, $\|Y_1\|_2$, and the norm of the $k$th block row $G_k(\alpha,\beta)$ of $E(\alpha,\beta)$. We find that

$$\|X_1\|_2 = \max(\|A_m\|_2, 1), \quad \|Y_1\|_2 \le m \max\Bigl(1, \max_{i=0:m-1} \|A_i\|_2\Bigr), \tag{3.9}$$

where we used Lemma 3.5 for $Y_1$. From Lemma 3.4 we have, for $j > 1$,

$$\|E(\alpha,\beta)_{ij}\|_2 \le \max_{\ell} \|A_\ell\|_2 \sum_{k=0}^{m-1} |\alpha|^k |\beta|^{m-k-1} = \|\Lambda_{\alpha,\beta}\|_1 \max_{\ell} \|A_\ell\|_2,$$

so that on using Lemma 3.5,

$$\|G_k(\alpha,\beta)\|_2 \le \sqrt{m}\, \|\Lambda_{\alpha,\beta}\|_1 \max(1, \max_i \|A_i\|_2), \tag{3.10}$$

this upper bound being independent of $k$. We can now bound the ratio $\eta_P(z_k,\alpha,\beta)/\eta_L(z,\alpha,\beta)$ in terms of the approximate right eigenpair $(z,\alpha,\beta)$ and the coefficient matrices defining $P$.

THEOREM 3.6. *Let $z$ be an approximate right eigenvector of $C_1$ corresponding to the approximate eigenvalue $(\alpha,\beta)$. Then for $z_k = z((k-1)n+1\colon kn)$, $k = 1\colon m$, we have*

$$\begin{aligned}\frac{1}{m^{1/2}} \le \frac{\eta_P(z_k,\alpha,\beta)}{\eta_{C_1}(z,\alpha,\beta)} &\le m^{3/2} \frac{(|\alpha|+|\beta|)\|\Lambda_{\alpha,\beta}\|_1 \max\bigl(1, \max_i \|A_i\|_2\bigr)^2}{\sum_{i=0}^{m} |\alpha|^i |\beta|^{m-i} \|A_i\|_2} \frac{\|z\|_2}{\|z_k\|_2} \\ &\le m^{5/2} \frac{\max\bigl(1, \max_i \|A_i\|_2\bigr)^2}{\min\bigl(\|A_0\|_2, \|A_m\|_2\bigr)} \frac{\|z\|_2}{\|z_k\|_2}.\end{aligned} \tag{3.11}$$

*Proof.* The first upper bound is obtained by combining (2.14) with (3.9) and (3.10). For the second upper bound it suffices to note that

$$\begin{aligned}\frac{(|\alpha|+|\beta|)\|\Lambda_{\alpha,\beta}\|_1}{\sum_{i=0}^{m} |\alpha|^i|\beta|^{m-i}\|A_i\|_2} &\le \frac{(|\alpha|+|\beta|)(|\alpha|^{m-1} + |\alpha|^{m-2}|\beta| + \cdots + |\beta|^{m-1})}{\min\bigl(\|A_0\|_2, \|A_m\|_2\bigr)(|\alpha|^m + |\beta|^m)} \\ &\le \frac{m}{\min\bigl(\|A_0\|_2, \|A_m\|_2\bigr)}\end{aligned}$$

by [7, Lem. A.1, (A.1)]. To prove the lower bound, let $\{\Delta A_i\}$ be an optimal set of perturbations in the definition of $\eta_P$. These trivially yield feasible perturbations

$\varDelta X_1 = \operatorname{diag}(\varDelta A_m, 0, \dots, 0)$ of $X_1$ and $\varDelta Y_1$ of $Y_1$, with $\varDelta Y_1$ being zero except for the first block row $[\varDelta A_{m-1}, \dots, \varDelta A_0]$. $\|\varDelta X_1\|_2 \le \eta_P \|X_1\|_2$ is immediate. Using Lemma 3.5,

$$\|\varDelta Y_1\|_2 \le m^{1/2} \max_{i=0:m-1} \|\varDelta A_i\|_2 \le m^{1/2}\eta_P \max_{i=0:m-1} \|A_i\|_2 \le m^{1/2}\eta_P \|Y_1\|_2. \qquad \square$$

The theorem reveals two main sufficient conditions for $\eta_P$ to be not much larger than $\eta_{C_1}$. The first is that $\|z\|_2/\|z_k\|_2$ is not much larger than 1. In the context of floating point arithmetic this requirement is to be expected, because if $\|z\|_2 \gg \|z_k\|_2$ then $z_k$ is likely to have suffered damaging subtractive cancellation in its formation. The second condition is that $\min(\|A_0\|_2, \|A_m\|_2) \approx \max_i \|A_i\|_2 \approx 1$, which is certainly true if $\|A_i\|_2 \approx 1$ for all $i$. Since $C_1 \in \mathbb{L}_1(P)$, Theorem 3.1 shows that the exact eigenvector is of the form $z = \Lambda_{\alpha,\beta} \otimes x$; since the largest element of $\Lambda_{\alpha,\beta}$ is the first or the last we can achieve $\|z\|_2/\|z_k\|_2 \in [1, \sqrt{m}]$ by taking $k = 1$ if $|\alpha| \ge |\beta|$ or $k = m$ if $|\alpha| \le |\beta|$. The importance for achieving a good backward error of recovering $x$ from the largest block component of $z$ has already been noted and shown empirically for the QEP by Tisseur [22, sec. 3.2]; our analysis provides theoretical confirmation for all degrees $m$.

We now turn to the backward error for a left eigenpair. Since $C_1 \in \mathbb{L}_1(P)$ with $v = e_1$ we have $L(\alpha,\beta)(\Lambda_{\alpha,\beta} \otimes I_n) = e_1 \otimes P(\alpha,\beta)$, so that (2.15) is satisfied with $H(\alpha,\beta) = \Lambda_{\alpha,\beta} \otimes I_n$ and $h = e_1$. The ensuing eigenvector recovery property is, from (2.17) or (3.5), $y = w(1\colon n)$. Before obtaining a backward error bound we give a more complete description of the relation between $y$ and $w$, which will aid in the interpretation of the bound. The following result extends [7, Lem. 7.2], which is stated for simple, finite, nonzero eigenvalues, to an arbitrary eigenvalue expressed in $(\alpha,\beta)$-form.

LEMMA 3.7 (left eigenvector recovery for $C_1$). *The vector $y \in \mathbb{C}^n$ is a left eigenvector of $P$ corresponding to the eigenvalue $(\alpha,\beta)$ if and only if*

$$w = \left\{ \begin{array}{ll} \begin{bmatrix} [\alpha^{m-1} I_n]^* \\ -[\alpha^{m-2}\beta A_{m-2} + \cdots + \alpha\beta^{m-2}A_1 + \beta^{m-1}A_0]^* \\ -[\alpha^{m-2}\beta A_{m-3} + \cdots + \alpha^2\beta^{m-3}A_1 + \alpha\beta^{m-2}A_0]^* \\ \vdots \\ -[\alpha^{m-2}\beta A_0]^* \end{bmatrix} y, & \alpha \ne 0, \\ \begin{bmatrix} [\beta^{m-1} I_n]^* \\ [\alpha\beta^{m-2}A_m + \beta^{m-1}A_{m-1}]^* \\ \vdots \\ [\alpha^{m-1}A_m + \cdots + \alpha\beta^{m-2}A_2 + \beta^{m-1}A_1]^* \end{bmatrix} y, & \beta \ne 0, \end{array} \right. \tag{3.12}$$

*is a left eigenvector of $C_1$ corresponding to $(\alpha,\beta)$. Every left eigenvector of $C_1$ with eigenvalue $(\alpha,\beta)$ is of the form* (3.12) *for some left eigenvector $y$ of $P$. For a finite eigenvalue, an alternative representation of $w$ is*

$$w^* = y^* \,[\, I_n, \quad B_{m-2}, \quad \dots \quad B_1, \quad B_0 \,],$$

*where $(P(t) - P(\lambda))/(t-\lambda) = \sum_{i=0}^{m-1} B_i t^i$ and $B_i = B_i(\lambda)$.*

*Proof.* Note first that the two different formulae in (3.12) (either of which can be obtained from the other by multiplying through by the conjugate of $(\alpha/\beta)^{m-1}$ or its reciprocal and using $y^*P(\alpha,\beta) = 0$) are needed because when $\alpha = 0$ (and hence

$y^*A_0 = 0$), the first expression is zero, while when $\beta = 0$ (and hence $y^*A_m = 0$), the second expression is zero.

For the first part it suffices to note that for $w$ as defined by (3.12) we have

$$w^*C_1(\alpha,\beta) = \begin{cases} y^*P(\alpha,\beta)(e_1^T \otimes I_n), & \alpha \neq 0, \\ y^*P(\alpha,\beta)(e_m^T \otimes I_n), & \alpha = 0. \end{cases}$$

For the next part, since $C_1$ is a strong linearization [4] and $P$ is regular, any eigenvalue $(\alpha,\beta)$ of $C_1$ of geometric multiplicity $k$ is also an eigenvalue of $P$ of geometric multiplicity $k$. Any $k$ linearly independent eigenvectors $y$ of $P$ for $(\alpha,\beta)$ clearly yield via (3.12) $k$ linearly independent eigenvectors of $L$. Hence any eigenvector of $L$ for $(\alpha,\beta)$ has the form (3.12).

The last part generalizes the analogous formula for scalar companion matrices given by Stewart [21, sec. 2]; we omit the proof. □

Lemma 3.7 shows that even when the eigenvalue is multiple all the left eigenvectors of $P$ can be obtained from the first $n$ components of the left eigenvectors of $C_1$.

We can now obtain the desired backward error bounds.

THEOREM 3.8. *Let $w$ be an approximate left eigenvector of $C_1$ corresponding to the approximate eigenvalue $(\alpha,\beta)$. Then for $w_1 = w(1\colon n)$ we have*

$$\begin{aligned} \frac{1}{m^{1/2}} \le \frac{\eta_P(w_1^*,\alpha,\beta)}{\eta_{C_1}(w^*,\alpha,\beta)} &\le m\frac{(|\alpha|+|\beta|)\|\Lambda_{\alpha,\beta}\|_2 \max(1,\max_i\|A_i\|_2)}{\sum_{i=0}^m |\alpha|^i|\beta|^{m-i}\|A_i\|_2}\frac{\|w\|_2}{\|w_1\|_2} \\ &\le m^{3/2}\frac{\max\bigl(1,\max_i\|A_i\|_2\bigr)}{\min(\|A_0\|_2,\|A_m\|_2)}\frac{\|w\|_2}{\|w_1\|_2}. \end{aligned} \tag{3.13}$$

*Proof.* The first upper bound follows directly from (2.18), (3.9), and $\|H_1(\alpha,\beta)\|_2 = \|\Lambda_{\alpha,\beta}\otimes I_n\|_2 = \|\Lambda_{\alpha,\beta}\|_2$. For the second upper bound it suffices to note that

$$\begin{aligned} \frac{(|\alpha|+|\beta|)\|\Lambda_{\alpha,\beta}\|_2}{\sum_{i=0}^m |\alpha|^i|\beta|^{m-i}\|A_i\|_2} &\le \frac{(|\alpha|+|\beta|)(|\alpha|^{2(m-1)}+|\alpha|^{2(m-2)}|\beta^2|+\cdots+|\beta|^{2(m-1)})^{1/2}}{\min\bigl(\|A_0\|_2,\|A_m\|_2\bigr)(|\alpha|^m+|\beta|^m)} \\ &\le \frac{m^{1/2}}{\min\bigl(\|A_0\|_2,\|A_m\|_2\bigr)} \end{aligned} \tag{3.14}$$

by [7, Lem. A.1, (A.3)]. The proof of the lower bound is exactly the same as in Theorem 3.6. □

Notice that compared with the bounds in Theorem 3.6 for right eigenpairs, the factor $\max_i\|A_i\|_2$ is not squared. However, $k$ is no longer a free parameter and so the ratio $\|w\|_2/\|w_1\|_2$ is fixed. Theorem 3.8 shows that $\eta_P(w_1^*,\alpha,\beta) \approx \eta_{C_1}(w^*,\alpha,\beta)$ is guaranteed provided that $\min(\|A_0\|_2,\|A_m\|_2) \approx \max_i\|A_i\|_2 \approx 1$ and $\|w\|_2/\|w_1\|_2$ is not much larger than 1. If $\|A_i\|_2 \lesssim 1$ for all $i$ then for an exact left eigenvector $w$ the ratio $\|w\|_2/\|w_1\|_2$ is bounded by about $(m^3/3)^{1/2}$; this can be seen from the first equation in (3.12) if $|\alpha| \ge |\beta|$ and the second if $|\alpha| \le |\beta|$.

A comparison with earlier work is instructive. Tisseur [22, Thm. 7] and Van Dooren and Dewilde [24, sec. 7] both show that solving a PEP by applying a backward stable solver to the first companion pencil is backward stable for the PEP, under certain conditions on the $A_i$. Van Dooren and Dewilde measure the perturbation $\Delta P$ to $P$ by $\|[\Delta A_m,\dots,\Delta A_0]\|_F/\|[A_m,\dots,A_0]\|_F$ and show that $\|[A_m,\dots,A_0]\|_F = 1$ implies stability. Tisseur uses the more stringent measure $\max_i\|\Delta A_i\|_2/\|A_i\|_2$, as in (2.1), and proves that $\|A_i\|_2 \equiv 1$ implies stability. These analyses are carried out without reference to specific eigenpairs or eigenvector recovery formulae and so they provide much less precise information than the bounds in Theorems 3.6 and 3.8.

**3.3. Scaled companion linearizations.** When the coefficient matrices of $P$ have norms that differ widely, the companion matrices $C_i(\lambda)$, $i = 1, 2$, are badly scaled and the bounds of Theorems 3.6 and 3.8 signal that $\eta_P \gg \eta_{C_1}$ is possible. In this section we study the effect on the backward error of scaling the identity blocks of $C_i$.

Let $D = \operatorname{diag}(d) \otimes I_n$, where $d \in \mathbb{R}^m$ with $d_1 = 1$ and $d_i > 0$, $i = 2{:}\,m$. It is easily checked that $DC_1(\lambda) \in \mathbb{L}_1(P)$ with $v = e_1$, that $C_2(\lambda)D \in \mathbb{L}_2(P)$ with $\widetilde{v} = e_1$, and that both scaled companion pencils are always linearizations. Since $C_2(P)D = (DC_1(P^T))^T$ we can concentrate on $DC_1$. The condition (2.11) becomes $G_k(\alpha,\beta)D^{-1} \cdot DC_1(\alpha,\beta) = e_k^T \otimes P(\alpha,\beta)$, where $G_k$ is defined in (3.8), and we find that

$$\|DX_1\|_2 = \max\Bigl(\max_{i>1} d_i,\, \|A_m\|_2\Bigr), \tag{3.15}$$

$$\|DY_1\|_2 \le m \max\Bigl(\max_{i>1} d_i,\, \max_{i=0:m-1} \|A_i\|_2\Bigr), \tag{3.16}$$

$$\|G_k(\alpha,\beta)D^{-1}\|_2 \le \sqrt{m}\,\|\Lambda_{\alpha,\beta}\|_1 \max\Bigl(1, \frac{\max_i \|A_i\|_2}{\min_{i>1} d_i}\Bigr). \tag{3.17}$$

In particular, if we choose $d_i = \max_\ell \|A_\ell\|_2$, $i = 2{:}\,m$, then

$$\|DX_1\|_2 = \max_i \|A_i\|_2, \qquad \|DY_1\|_2 \le m \max_i \|A_i\|_2,$$
$$\|G_k(\alpha,\beta)D^{-1}\|_2 \le \sqrt{m}\,\|\Lambda_{\alpha,\beta}\|_1.$$

As we now show, this scaling yields bounds for $\eta_P/\eta_{DC_1}$ better than those for $\eta_P/\eta_{C_1}$. We introduce the quantity

$$\rho = \frac{\max_i \|A_i\|_2}{\min(\|A_0\|_2, \|A_m\|_2)}, \tag{3.18}$$

which measures the scaling of the problem.

THEOREM 3.9. *Let* $D_s = \operatorname{diag}(1, s, \dots, s) \otimes I_n \in \mathbb{R}^{mn \times mn}$ *with* $s = \max_i \|A_i\|_2$. *Let* $z$ *and* $w$ *be approximate right and left eigenvectors of* $D_sC_1$ *corresponding to the approximate eigenvalue* $(\alpha,\beta)$. *Then for* $z_k = z((k-1)n + 1{:}\,kn)$, $k = 1{:}\,m$, *we have*

$$\frac{1}{m^{1/2}} \le \frac{\eta_P(z_k,\alpha,\beta)}{\eta_{D_sC_1}(z,\alpha,\beta)} \le m^{3/2}\, \frac{(|\alpha| + |\beta|)\|\Lambda_{\alpha,\beta}\|_1 \max_i \|A_i\|_2}{\sum_{i=0}^m |\alpha|^i|\beta|^{m-i}\|A_i\|_2}\, \frac{\|z\|_2}{\|z_k\|_2} \le m^{5/2}\,\rho \frac{\|z\|_2}{\|z_k\|_2},$$

*and for* $w_1 = w(1{:}\,n)$,

$$\frac{1}{m^{1/2}} \le \frac{\eta_P(w_1^*,\alpha,\beta)}{\eta_{D_sC_1}(w^*,\alpha,\beta)} \le m\, \frac{(|\alpha| + |\beta|)\|\Lambda_{\alpha,\beta}\|_2 \max_i \|A_i\|_2}{\sum_{i=0}^m |\alpha|^i|\beta|^{m-i}\|A_i\|_2}\, \frac{\|w\|_2}{\|w_1\|_2} \le m^{3/2}\,\rho \frac{\|w\|_2}{\|w_1\|_2}.$$

*Proof.* The proof is analogous to the proofs of Theorems 3.6 and 3.8, making use of (3.15)–(3.17). □

The bounds of Theorem 3.9 for the scaled companion pencil improve upon those for the unscaled pencil in several ways.

1. For the right eigenvector, the term $\max(1, \max_i \|A_i\|_2^2)/\min(\|A_0\|_2, \|A_m\|_2)$ in (3.11) is replaced by $\rho$, which is much smaller if $\max_i \|A_i\|_2 \gg 1$ or $\max_i \|A_i\|_2 \ll 1$.
2. For the left eigenvector, the term $\max(1, \max_i \|A_i\|_2)/\min(\|A_0\|_2, \|A_m\|_2)$ in (3.13) is replaced by $\rho$, which is much smaller if $\max_i \|A_i\|_2 \ll 1$.

3. For the scaled companion pencil, $\|w\|_2/\|w_1\|_2$ is guaranteed to be $O(m^{3/2})$ for the exact eigenvector, as can be seen from the appropriate choice of formula in (3.12), bearing in mind that scaling changes $w$ in (3.12) to $D_s^{-1}w$. To draw the same conclusion for the unscaled pencil we require $\max_i \|A_i\|_2 \lesssim 1$.

Our bounds suggest that scaling the identity blocks of $C_1$ can significantly improve the backward error of the recovered eigenvectors of $P$. We can, of course, employ more sophisticated two-sided scalings, including balancing [16], [25]. However, these scalings produce a new pencil not belonging to $\mathbb{L}_1$, so our backward error bounds are not applicable to them.

**3.4. $\mathbb{DL}(P)$ linearizations.** From section 2.2 and the definition of $\mathbb{L}_1$ in (3.3), it is clear that for pencils $L \in \mathbb{L}_1$ our analysis provides upper bounds for the backward error $\eta_P$ associated with approximate *left* eigenvectors of $P$ recovered from approximate *left* eigenvectors of $L$. The same is true for $L \in \mathbb{L}_2$ and approximate *right* eigenvectors. We now concentrate on the intersection

$$\mathbb{DL}(P) = \mathbb{L}_1(P) \cap \mathbb{L}_2(P), \tag{3.19}$$

since for pencils in $\mathbb{DL}(P)$ we can obtain backward error bounds for both left *and* right eigenvectors. $\mathbb{DL}(P)$ is a much smaller space than $\mathbb{L}_1(P)$ and $\mathbb{L}_2(P)$, being just $m$-dimensional. Indeed, it is shown in [19, Thm. 5.3] and [6, Thm. 3.4] that $L \in \mathbb{DL}(P)$ if and only if $L$ satisfies the conditions in (3.3) and (3.4) with $\widetilde{v} = v$. The general form of $\mathbb{DL}(P)$ for the quadratic $P(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0$ is given by

$$\mathbb{DL}(P) = \left\{ L(\lambda) = \lambda \begin{bmatrix} v_1A_2 & v_2A_2 \\ v_2A_2 & v_2A_1 - v_1A_0 \end{bmatrix} + \begin{bmatrix} v_1A_1 - v_2A_2 & v_1A_0 \\ v_1A_0 & v_2A_0 \end{bmatrix} : \ v \in \mathbb{C}^2 \right\},$$

which illustrates the fact that the companion pencils are not contained in $\mathbb{DL}(P)$ for any $m$. Just as for $\mathbb{L}_1$ and $\mathbb{L}_2$, almost all pencils in $\mathbb{DL}(P)$ are linearizations [19, Thm. 6.8]. In fact, there is a beautiful characterization of the subset of pencils $L \in \mathbb{DL}(P)$ that are linearizations [19, Thm. 6.7]: they are those for which no eigenvalue of $P$ is a root of the polynomial $\mathsf{p}(\lambda; v) := v^T \Lambda = \sum_{i=1}^m v_i \lambda^{m-i}$, where when $v_1 = 0$ we define $\infty$ to be a root of $\mathsf{p}(\lambda; v)$. Throughout this section we assume that the pencils $L \in \mathbb{DL}(P)$ under consideration are linearizations.

For pencils $L \in \mathbb{DL}(P)$, we have, by definition,

$$L(\alpha,\beta)(\Lambda_{\alpha,\beta} \otimes I_n) = v \otimes P(\alpha,\beta), \quad (\Lambda_{\alpha,\beta}^T \otimes I_n)L(\alpha,\beta) = v^T \otimes P(\alpha,\beta),$$

so that (2.11) and (2.15) hold with $G(\alpha,\beta) = \Lambda_{\alpha,\beta}^T \otimes I_n$ and $H(\alpha,\beta) = \Lambda_{\alpha,\beta} \otimes I_n$. Moreover, $\|G(\alpha,\beta)\|_2 = \|H(\alpha,\beta)\|_2 = \|\Lambda_{\alpha,\beta}\|_2$. From [7, Lem. 4.1] we know that $L(\alpha,\beta) = \alpha X + \beta Y$ satisfies

$$\max\bigl(\|X\|_2, \|Y\|_2\bigr) \le m r^{1/2} \max_i \|A_i\|_2, \tag{3.20}$$

where $r$ is the number of nonzeros in $v$ and we assume $\|v\|_2 = 1$ without loss of generality. We now have the ingredients to obtain a backward error bound. Recall that $\rho$ is defined in (3.18).

THEOREM 3.10. *Let $L \in \mathbb{DL}(P)$ with vector $v$ in* (3.3) *be a linearization, where $v$ has unit* 2*-norm and $r$ nonzeros. Let $z$ be an approximate right eigenvector of $L$ corresponding to the approximate eigenvalue $(\alpha,\beta)$. Then for $x = \sum_{i=1}^m v_i z_i$, where $z_i = z((i-1)n+1\colon in)$, we have*

$$\frac{\eta_P(x,\alpha,\beta)}{\eta_L(z,\alpha,\beta)} \le m r^{1/2} \frac{(|\alpha|+|\beta|)\|\Lambda_{\alpha,\beta}\|_2 \max_i \|A_i\|_2}{\sum_{i=0}^m |\alpha|^i |\beta|^{m-i} \|A_i\|_2} \frac{\|z\|_2}{\|x\|_2} \le m^{3/2} r^{1/2} \rho \frac{\|z\|_2}{\|x\|_2}.$$

*Proof.* Combine (2.14), (3.20), and (3.14). ☐

Note that the exact $z$ has the form $\Lambda_{\alpha,\beta} \otimes \xi$ so that $\|z\|_2 = \|\Lambda_{\alpha,\beta}\|_2 \|\xi\|_2$, and $\|x\|_2 = \| \sum_{i=1}^m v_i z_i \|_2 = |\Lambda_{\alpha,\beta}^T v| \|\xi\|_2$. Hence $\|z\|_2 / \|x\|_2 = \|\Lambda_{\alpha,\beta}\|_2 / |\mathsf{p}(\alpha,\beta;v)|$, where $\mathsf{p}(\alpha,\beta;v) = \Lambda_{\alpha,\beta}^T v = \sum_{i=1}^m v_i \alpha^{m-i} \beta^{i-1}$. Thus $\min\{ \|z\|_2/\|x\|_2 : \|v\|_2 = 1\} = 1$, with equality attained for $v_* = \overline{\Lambda_{\alpha,\beta}} / \|\Lambda_{\alpha,\beta}\|_2$. This choice of $v$ minimizes the second upper bound of Theorem 3.10. However, simply choosing $v = e_k$ where $\|z_k\|_2 = \max_{i=1:m} \|z_i\|_2$ ensures that $\|z\|_2/\|x\|_2 \le \sqrt{m}$, which is perfectly adequate.

Intuitively, we might expect that $\eta_P(x,\alpha,\beta) \ge \eta_L(z,\alpha,\beta)$, at least to within some constant factor, but this is not necessarily the case. Consider, for example, the pencil

$$L(\lambda) = \lambda \begin{bmatrix} A & A \\ A & B - C \end{bmatrix} + \begin{bmatrix} B - A & C \\ C & C \end{bmatrix} \in \mathbb{DL}(\lambda^2 A + \lambda B + C),$$

which corresponds to $v = [1\ 1]^T$ in (3.3). Suppose $A = B = I$ and $C = \epsilon I$ with $0 < \epsilon \ll 1$, and let $\Delta A = \delta I$ and $\Delta B = \Delta C = 0$. These perturbations have relative size $\max(\|\Delta A\|_2/\|A\|_2, \|\Delta B\|_2/\|B\|_2, \|\Delta C\|_2/\|C\|_2) = \delta$, but for the pencil $\max(\|\Delta X\|_2/\|X\|_2, \|\Delta Y\|_2/\|Y\|_2) \approx \max(\delta, \delta/\epsilon) = \delta/\epsilon$. Hence a small perturbation to $P$ does not necessarily correspond to a small perturbation to $L$ and $\eta_P/\eta_L$ cannot therefore be bounded below by a positive constant. This phenomenon is not present for the pencils corresponding to $v = e_k$, which form the standard basis for $\mathbb{DL}(P)$ [6], because for these pencils each block of $X$ and $Y$ is plus or minus a single block $A_i$. We now specialize Theorem 3.10 to these pencils.

COROLLARY 3.11. *Let $L_k \in \mathbb{DL}(P)$ corresponding to $v = e_k$ in* (3.3) *be a linearization. Let $z$ be an approximate right eigenvector of $L_k$ corresponding to the approximate eigenvalue $(\alpha,\beta)$. Then for $z_k = z((k-1)n + 1{:}\,kn)$, we have*

$$\frac{1}{m} \le \frac{\eta_P(z_k,\alpha,\beta)}{\eta_{L_k}(z,\alpha,\beta)} \le m \frac{(|\alpha| + |\beta|)\|\Lambda_{\alpha,\beta}\|_2 \max_i \|A_i\|_2}{\sum_{i=0}^m |\alpha|^i |\beta|^{m-i} \|A_i\|_2} \frac{\|z\|_2}{\|z_k\|_2} \le m^{3/2} \rho \frac{\|z\|_2}{\|z_k\|_2}.$$

*Proof.* The upper bound follows from Theorem 3.10. The lower bound is proved in a similar way to the lower bound of Theorem 3.6. ☐

Analogues of Theorem 3.10 and Corollary 3.11 hold for approximate left eigenvectors $w$ of $L_k$: $z$ is simply replaced by $w$ and $x$ by $y = \sum_{i=1}^m \overline{v}_i w_i$.

With the notation in Corollary 3.11, the exact eigenvector $z$ satisfies $z = \Lambda_{\alpha,\beta} \otimes x$, and it is easy to see that $\|z\|_2/\|z_k\|_2 \approx 1$ for $k = 1$ if $|\alpha| \ge |\beta|$ and for $k = m$ if $|\alpha| \le |\beta|$. Assuming the approximate eigenvector $z$ shares the latter property, the pencils in $\mathbb{DL}(P)$ with $v = e_1$ and $v = e_m$ yield backward errors $\eta_P \approx \eta_L$ for eigenpairs with eigenvectors of modulus greater than or less than 1, respectively, provided that the measure $\rho$ of the scaling of the problem is of order 1. Two points are worth noting.

1. Although an eigenvector of $P$ can be recovered from any of the blocks $z_i = z((i-1)n + 1{:}\,in)$ of an eigenvector $z$ of $L_k$ (see Theorem 3.1), our backward error bounds in Corollary 3.11 require $i = k$.
2. The pencils $L_1$ and $L_m$ are indeed linearizations if $A_0$ and $A_m$, respectively, are nonsingular, as can be seen from the characterization mentioned at the start of this subsection.

### 3.5. Comparison with conditioning results.

Backward error and conditioning are complementary concepts. Ideally, we would like the linearization $L$ that we use to be as well conditioned as the original polynomial $P$ and for it to lead, after recovering an approximate eigenpair of $P$ from one of $L$, to a backward error $\eta_P$

of the same order of magnitude as $\eta_L$. Therefore to show that one linearization is preferable to another we need to show that it enjoys a better bound for $\eta_P/\eta_L$ as well as a better condition number bound. Remarkably, our backward error results are entirely harmonious with the results of Higham, Mackey, and Tisseur [7] concerning eigenvalue conditioning, as we now explain.

For the companion forms the analysis in [7, sec. 7] provides bounds on the ratio $\kappa_{C_1}(\lambda)/\kappa_P(\lambda)$ of appropriately defined condition numbers in the case of quadratics. That analysis is readily extended to $(\alpha,\beta)$-form and general degrees, and it shows that, like the backward error ratio in Theorem 3.6, $\kappa_{C_1}(\alpha,\beta)/\kappa_P(\alpha,\beta)$ is bounded by a multiple of $\max\bigl(1, \max_i \|A_i\|_2^2\bigr)/\min(\|A_0\|_2, \|A_m\|_2)$. Thus if $\min(\|A_0\|_2, \|A_m\|_2) \approx \max_i \|A_i\|_2 \approx 1$ and if a relatively large block is used for right eigenvector recovery then $C_1$ is an optimal linearization from the points of view of both backward error and conditioning.

For the scaled companion forms we can show that $\kappa_{D_sC_1}(\alpha,\beta)/\kappa_P(\alpha,\beta)$ is bounded by a multiple of $\rho$, just as for the backward error ratios in Theorem 3.9. So if $\rho \approx 1$ and a relatively large block is used for eigenvector recovery then $D_sC_1$ is an optimal linearization.

For the $\mathbb{DL}(P)$ pencils with $v = e_1$ (if $|\lambda| \ge 1$) or $v = e_m$ (if $|\lambda| \le 1$) it is once again the case that the factor $\rho$ in the backward error bound (in Corollary 3.11) is also the key quantity in a bound on the ratio of condition numbers $\kappa_{L_k}/\kappa_P$. We can conclude that if $\rho \approx 1$ then $L_1$ and $L_m$ are optimal with respect to both backward error and conditioning over all linearizations for $|\lambda|$ greater than 1 and less than 1, respectively, assuming $\|z_1\|_2 \approx \|z\|_2$ for $L_1$ and $\|z_m\|_2 \approx \|z\|_2$ for $L_m$ (properties that hold for the exact eigenvectors).

## 4. Structured linearizations.

We now briefly consider to what extent the results above extend to structured linearizations for structured polynomials. Our definition of backward error remains the same and so does not incorporate structure. The issue is that structure may change some key properties of a linearization and thereby may limit our freedom in choosing how to recover eigenvectors.

### 4.1. Symmetric and Hermitian structures.

If $P$ is symmetric, that is, $P(\lambda) = P(\lambda)^T$, then all the pencils in $\mathbb{DL}(P)$ are symmetric, and these comprise all the symmetric pencils in $\mathbb{L}_1(P)$ [6, Thm 5.2]. Hence Theorem 3.10 and Corollary 3.11 are both applicable with $L$ symmetric. If $P$ is Hermitian, that is, $P(\lambda) = P(\bar{\lambda})^*$, then it is precisely the pencils in $\mathbb{DL}(P)$ with a *real* vector $v$ that are Hermitian [6, Thm 6.2]. Theorem 3.10 remains applicable for Hermitian $L$ with the minor restriction that $v$ is real. Thus symmetry and Hermitian structure impose no significant limitations on the applicability of our backward error bounds.

### 4.2. Alternating and palindromic structures.

We now consider some other classes of structures for which we can identify structured linearizations. These structures are less familiar than symmetric or Hermitian structures but still important in a variety of applications [17, Chap. 7]. In what follows, the symbol $\star$ is used as an abbreviation for transpose ($T$) in the real case and either transpose or conjugate transpose ($*$) in the complex case. The $\star$-adjoint of $P$ is defined by

$$P^\star(\lambda) = \sum_{i=0}^{m} \lambda^i A_i^\star.$$

$P(\lambda)$ is said to be

$\star$-even if $P^\star(-\lambda) = P(\lambda)$, $\star$-odd if $P^\star(-\lambda) = -P(\lambda)$,
$\star$-palindromic if $\mathrm{rev}P^\star(\lambda) = P(\lambda)$, $\star$-antipalindromic if $\mathrm{rev}P^\star(\lambda) = -P(\lambda)$,
where rev is defined in (3.2). For example, the quadratic $Q(\lambda) = \lambda^2 M + \lambda G + K$ with $M$, $K$ symmetric and $G$ skew-symmetric, arising in gyroscopic systems, is $T$-even since $Q^T(-\lambda) = Q(\lambda)$. On the other hand, the quadratic $Q(\lambda) = \lambda^2 A + \lambda B + A^T$ with $B$ complex symmetric, arising in the study of vibration of rail tracks under the excitation of high speed trains [10], [11], is $T$-palindromic since $\mathrm{rev}Q^T(\lambda) = Q(\lambda)$.

Linearizations in $\mathbb{L}_1(P)$ that reflect the structure of these polynomials and therefore preserve symmetries in their spectra have recently been investigated by Mackey et al. [18]. It is shown in [18, Thms. 3.5, 3.6] that if $L(\lambda) \in \mathbb{L}_1(P)$ is $\star$-structured with vector $v$ then $(M \otimes I_n)L(\lambda)$ is in $\mathbb{DL}(P)$ with vector $Mv$, where $M$ is either a diagonal matrix of alternating signs, $M = \mathrm{diag}((-1)^{m-1}, \dots, (-1)^0)$, in the case of even/odd structures, or the reverse identity matrix, $R = (\delta_{i,n+1-i})$, in the context of palindromic structures.

Since $L$ itself is in general not in $\mathbb{DL}(P)$ we cannot apply Theorem 3.10. However, the proof of the theorem is readily adapted, and by exploiting the fact that $M \otimes I_n$ is unitary the same bound is obtained.

THEOREM 4.1. *Let $L \in \mathbb{L}_1(P)$ with vector $v$ be a $\star$-structured linearization and assume that $v$ has unit 2-norm and $r$ nonzeros. Let $z$ be an approximate right eigenvector of $L$ corresponding to the approximate eigenvalue $(\alpha, \beta)$. Then for $x = \sum_{i=1}^m v_i z_i$ we have*

$$\frac{\eta_P(x,\alpha,\beta)}{\eta_L(z,\alpha,\beta)} \le m r^{1/2} \frac{(|\alpha| + |\beta|)\|\Lambda_{\alpha,\beta}\|_2 \max_i \|A_i\|_2}{\sum_{i=0}^m |\alpha|^i |\beta|^{m-i} \|A_i\|_2} \frac{\|z\|_2}{\|x\|_2} \le m^{3/2} r^{1/2} \rho \frac{\|z\|_2}{\|x\|_2}.$$

*For approximate left eigenvectors an analogous bound holds with $z$ replaced by $w$ and $x$ by $y = \sum_{i=1}^m (M\overline{v})_i w_i$.*

Theorem 4.1 shows that $\eta_P \approx \eta_L$ as long as $\rho = O(1)$ and $\|z\|_2/\|x\|_2 \approx 1$. However, whereas for $\mathbb{DL}(P)$ $v$ can be freely chosen, in particular to minimize $\|z\|_2/\|x\|_2$, now the choice of $v$ is constrained by the requirement that $L$ be $\star$-structured. For example, for $T$-palindromic polynomials $P$, $L \in \mathbb{L}_1(P)$ with vector $v$ is $T$-palindromic if and only $Rv = v$ [18, Thm. 3.5]; in the case of a quadratic, $v = [1\ 1]/\sqrt{2}$ is forced.

## 5. Quadratic polynomials.

We now concentrate our attention on quadratic polynomials, $Q(\lambda) = \lambda^2 A + \lambda B + C$, for which we can give a more detailed analysis than in the general case, covering in particular a potentially very beneficial scaling of the polynomial. We write

$$a = \|A\|_2, \quad b = \|B\|_2, \quad c = \|C\|_2. \tag{5.1}$$

Note that $\Lambda_{\alpha,\beta} = [\alpha, \beta]^T$. We will recover eigenvectors of $Q$ from the components $z_1 = z(1\colon n)$ and $z_2 = z(n+1\colon 2n)$ (and similarly for $w$) of eigenvectors of a linearization.

The first companion form of $Q$ is given by

$$C_1(\lambda) = \lambda \begin{bmatrix} A & 0 \\ 0 & I_n \end{bmatrix} + \begin{bmatrix} B & C \\ -I_n & 0 \end{bmatrix},$$

and $D_s C_1(\lambda) = \mathrm{diag}(I_n, sI_n) C_1(\lambda)$ with $s = \max(a,b,c)$. We normalize so that $|\alpha|^2 + |\beta|^2 = 1$. Theorems 3.6 and 3.9 say that for right eigenpairs

$$\frac{\eta_Q(z_k,\alpha,\beta)}{\eta_{C_1}(z,\alpha,\beta)} \le 2^{5/2} \frac{\max\left(1,a,b,c\right)^2}{|\alpha|^2 a + |\alpha||\beta| b + |\beta|^2 c} \frac{\|z\|_2}{\|z_k\|_2}, \quad k = 1, 2, \tag{5.2}$$

$$\frac{\eta_Q(z_k,\alpha,\beta)}{\eta_{D_sC_1}(z,\alpha,\beta)} \le 2^{5/2}\frac{\max(a,b,c)}{|\alpha|^2a+|\alpha||\beta|b+|\beta|^2c}\frac{\|z\|_2}{\|z_k\|_2}, \quad k=1,2. \tag{5.3}$$

Analogous bounds hold for left eigenvectors: they have factor $2^{3/2}$ and there is no square in the numerator for the analogue of (5.2). In interpreting these bounds and those below recall that, for the exact eigenvectors of any pencil in $\mathbb{L}_1(Q)$,

$$\frac{\|z\|_2}{\|z_1\|_2} \approx 1 \quad \text{for } |\alpha| \ge |\beta|, \qquad \frac{\|z\|_2}{\|z_2\|_2} \approx 1 \quad \text{for } |\alpha| \le |\beta|. \tag{5.4}$$

The $\mathbb{DL}(Q)$ pencils with $v = e_1$ and $v = e_2$ are given by

$$L_1(\lambda) = \lambda\begin{bmatrix} A & 0 \\ 0 & -C\end{bmatrix} + \begin{bmatrix} B & C \\ C & 0\end{bmatrix}, \quad L_2(\lambda) = \lambda\begin{bmatrix} 0 & A \\ A & B\end{bmatrix} + \begin{bmatrix} -A & 0 \\ 0 & C\end{bmatrix}.$$

We know from Corollary 3.11 that

$$\frac{\eta_Q(z_1,\alpha,\beta)}{\eta_{L_1}(z,\alpha,\beta)} \le 2^{3/2}\frac{\max(a,b,c)}{|\alpha|^2a+|\alpha||\beta|b+|\beta|^2c}\frac{\|z\|_2}{\|z_1\|_2}. \tag{5.5}$$

In view of (5.4) this bound is appropriate when $|\alpha| \ge |\beta|$. If $|\alpha| \le |\beta|$ then we wish to take $z_2$ rather than $z_1$ as eigenvector of $Q$, but Theorem 3.10 does not provide a bound for $L_1$ and $z_2$. We now derive such a bound, by explicitly constructing an appropriate $G$ matrix. It is easy to check that $G_Q(\alpha,\beta) = [\,\beta I_n, \quad -(\alpha A+\beta B)C^{-1}\,]$ satisfies $G_Q(\alpha,\beta)L_1(\alpha,\beta) = e_2^T \otimes Q(\alpha,\beta)$ so that (2.11) holds, and by Lemma 3.5

$$\|G_Q(\alpha,\beta)\|_2 \le \sqrt{2}\,\|\Lambda_{\alpha,\beta}\|_\infty \max\bigl(1,(a+b)\|C^{-1}\|_2\bigr).$$

Hence (2.14) yields

$$\frac{\eta_Q(z_2,\alpha,\beta)}{\eta_{L_1}(z,\alpha,\beta)} \le 4\frac{\max(a,b,c)\max(1,(a+b)\|C^{-1}\|_2)}{|\alpha|^2a+|\alpha||\beta|b+|\beta|^2c}\frac{\|z\|_2}{\|z_2\|_2}. \tag{5.6}$$

Similarly we have for $L_2$, by an analogue of the $G_Q$ analysis and by Corollary 3.11,

$$\frac{\eta_Q(z_1,\alpha,\beta)}{\eta_{L_2}(z,\alpha,\beta)} \le 4\frac{\max(a,b,c)\max(1,(b+c)\|A^{-1}\|_2)}{|\alpha|^2a+|\alpha||\beta|b+|\beta|^2c}\frac{\|z\|_2}{\|z_1\|_2}, \tag{5.7}$$

$$\frac{\eta_Q(z_2,\alpha,\beta)}{\eta_{L_2}(z,\alpha,\beta)} \le 2^{3/2}\frac{\max(a,b,c)}{|\alpha|^2a+|\alpha||\beta|b+|\beta|^2c}\frac{\|z\|_2}{\|z_2\|_2}. \tag{5.8}$$

Essentially the same bounds (5.5)–(5.8) hold for approximate left eigenvectors: $z$ is simply replaced by $w$ and $z_i$ by $w_i$.

In Table 5.1 we summarize for unstructured quadratics the main conclusions from these bounds concerning conditions that guarantee $\eta_P \approx \eta_L$. Here, using $\rho$ from (3.18),

$$\rho = \frac{\max(a,b,c)}{\min(a,c)} \ge \frac{\max(a,b,c)}{|\alpha|^2a+|\alpha||\beta|b+|\beta|^2c}. \tag{5.9}$$

In view of the bounds, it is natural to scale the problem to try to bring the 2-norms of $A$, $B$, and $C$ close to 1. The scaling of Fan, Lin, and Van Dooren [2] has

Table 5.1

*Sufficient conditions for* $\eta_P \approx \eta_L$; $\rho$ *is defined in* (5.9).

| Linearization | Eigenvalue | Right eigenvector | Left eigenvector | Condition |
|---|---|---|---|---|
| Companion | $\lvert\alpha\rvert \ge \lvert\beta\rvert$<br>$\lvert\alpha\rvert \le \lvert\beta\rvert$ | $z_1$<br>$z_2$ | $w_1$ | $b \le a \approx c \approx 1$ |
| Scaled companion | $\lvert\alpha\rvert \ge \lvert\beta\rvert$<br>$\lvert\alpha\rvert \le \lvert\beta\rvert$ | $z_1$<br>$z_2$ | $w_1$ | $\rho \approx 1$ |
| $L_1$ | $\lvert\alpha\rvert \ge \lvert\beta\rvert$<br>$\lvert\alpha\rvert \le \lvert\beta\rvert$ | $z_1$<br>$z_2$ | $w_1$<br>$w_2$ | $\rho \approx 1$<br>$\rho \max\bigl(1, (a+b)\lVert C^{-1}\rVert_2\bigr) \approx 1$ |
| $L_2$ | $\lvert\alpha\rvert \ge \lvert\beta\rvert$<br>$\lvert\alpha\rvert \le \lvert\beta\rvert$ | $z_1$<br>$z_2$ | $w_1$<br>$w_2$ | $\rho \max\bigl(1, (b+c)\lVert A^{-1}\rVert_2\bigr) \approx 1$<br>$\rho \approx 1$ |

precisely this aim. It converts $Q(\lambda) = \lambda^2 A + \lambda B + C$ to $\widetilde{Q}(\mu) = \mu^2 \widetilde{A} + \mu \widetilde{B} + \widetilde{C}$, where

$$\lambda = \gamma\mu, \quad Q(\lambda)\delta = \mu^2(\gamma^2\delta A) + \mu(\gamma\delta B) + \delta C \equiv \widetilde{Q}(\mu), \tag{5.10a}$$

$$\gamma = \sqrt{c/a}, \quad \delta = 2/(c + b\gamma). \tag{5.10b}$$

Letting

$$\widetilde{a} = \lVert\widetilde{A}\rVert_2, \quad \widetilde{b} = \lVert\widetilde{B}\rVert_2, \quad \widetilde{c} = \lVert\widetilde{C}\rVert_2, \tag{5.11}$$

$$\tau = \frac{b}{\sqrt{ac}}, \tag{5.12}$$

we have

$$\widetilde{a} = \widetilde{c} = \frac{2}{1+\tau}, \qquad \widetilde{b} = \frac{2\tau}{1+\tau}, \qquad \frac{\widetilde{a}}{2} + \widetilde{b} + \frac{\widetilde{c}}{2} = 2,$$

so that $2/3 \le \max(\widetilde{a}, \widetilde{b}, \widetilde{c}) \le 2$. It is straightforward to show that $\widetilde{\rho} = \max(\widetilde{a}, \widetilde{b}, \widetilde{c})/\min(\widetilde{a}, \widetilde{c}) = \max(1, \tau) \le \rho$. Note that $\eta_Q(x, \lambda) = \eta_{\widetilde{Q}}(x, \mu)$, so this scaling has no effect on the backward error for the quadratic; its purpose is to improve the backward error for the linearization. For $\widetilde{Q}(\mu)$, the bounds (5.2), (5.3), and (5.5)–(5.8) can be simplified.

Theorem 5.1. *Let* $(z, w, \alpha, \beta)$ *be an approximate eigentriple of a linearization of the scaled quadratic* $\widetilde{Q}$ *in* (5.10) *with* $|\alpha|^2 + |\beta|^2 = 1$. *Define*

$$\omega = \omega(\alpha, \beta) := \frac{1+\tau}{1 + |\alpha\beta|\tau}, \tag{5.13}$$

*with* $\tau$ *as in* (5.12). *We have*

$$\frac{\eta_{\widetilde{Q}}(z_i, \alpha, \beta)}{\eta_{C_1}(z, \alpha, \beta)} \le 2^{7/2}\omega \frac{\lVert z\rVert_2}{\lVert z_i\rVert_2}, \quad i = 1, 2, \qquad \frac{\eta_{\widetilde{Q}}(w_1^*, \alpha, \beta)}{\eta_{C_1}(w^*, \alpha, \beta)} \le 2^{3/2}\omega \frac{\lVert w\rVert_2}{\lVert w_1\rVert_2}.$$

*The same bounds hold for* $D_s C_1$ *and the constant* $2^{7/2}$ *can be replaced by* $2^{5/2}$. *Furthermore,*

$$\frac{\eta_{\widetilde{Q}}(x, \alpha, \beta)}{\eta_{L_i}(z, \alpha, \beta)} \le f_i(x)\omega \frac{\lVert z\rVert_2}{\lVert x\rVert_2}, \quad i = 1, 2,$$

*with*

$$f_1(x) = \begin{cases} 2^{3/2} & \textit{if } x = z_1, \\ 8\|\widetilde{C}^{-1}\|_2 & \textit{if } x = z_2, \end{cases} \quad f_2(x) = \begin{cases} 8\|\widetilde{A}^{-1}\|_2 & \textit{if } x = z_1, \\ 2^{3/2} & \textit{if } x = z_2, \end{cases}$$

*where the nonsingularity of $C$ and $A$ is required for $f_1(z_2)$ and $f_2(z_1)$. Similar bounds hold for approximate left eigenvectors and $L_i$, $i = 1, 2$: $z$ is replaced by $w$ and $z_i$ by $w_i$.*

*Proof.* For the scaled norms in (5.11) we have

$$(5.14)\quad |\alpha|^2\widetilde{a} + |\alpha||\beta|\widetilde{b} + |\beta|^2\widetilde{c} = \frac{2}{1+\tau} + |\alpha||\beta|\frac{2\tau}{1+\tau} = \frac{2(1+|\alpha||\beta|\tau)}{1+\tau} = \frac{2}{\omega(\alpha,\beta)}$$

and the upper bounds follow from (5.2), (5.3), and (5.5)–(5.8). ☐

We can regard $\omega$ in (5.13) as a growth factor bound in the translation from backward error for $L$ to backward error for $\widetilde{Q}$. With the normalization $|\alpha|^2 + |\beta|^2 = 1$, which implies $|\alpha||\beta| \le 1/2$, this factor satisfies the bounds

$$(5.15)\qquad 1 \le \frac{1+\tau}{1+\frac{1}{2}\tau} \le \omega \le \min\left\{1+\tau, \frac{1}{|\alpha\beta|}\right\} \le 1+\tau.$$

Fan, Lin, and Van Dooren [2] identify $\max(1+\tau, 1+\tau^{-1})$ as a growth factor. Our bounds for $\omega$, which unlike in [2] are for individual eigenpairs and apply to $L_i$ as well as $C_1$, are sharper in two respects. First, they show that $\tau$ satisfying $\tau \ll 1$ are harmless, since our upper bound for $\omega$ is $O(1)$. Second, even when $\tau \gg 1$ the penultimate bound in (5.15) will still be of order 1 if $|\alpha||\beta| = |\alpha|\sqrt{1-|\alpha|^2} = O(1)$, which is the case unless $|\lambda| = |\alpha|/|\beta| = |\alpha|/\sqrt{1-|\alpha|^2}$ is small or large.

The most striking consequence of the theorem is that if

$$(5.16)\qquad \|B\|_2 \lesssim (\|A\|_2\|C\|_2)^{1/2},$$

so that $\tau = O(1)$ and hence $\omega = O(1)$, then the $\eta_{Q}/\eta_L$ ratios are 1 for the relevant choice of $z_i$, provided $\widetilde{A}^{-1}$ and $\widetilde{C}^{-1}$ have norms of order 1 in the case of two of the bounds for $L_1$ and $L_2$. In the terminology of quadratics arising from mechanical systems with damping, the condition (5.16) holds for systems that are not too heavily damped. A class of problems for which (5.16) is satisfied is the elliptic $Q$ [9], [13]: those for which $A$ is Hermitian positive definite, $B$ and $C$ are Hermitian, and $(x^*Bx)^2 < 4(x^*Ax)(x^*Cx)$ for all nonzero $x \in \mathbb{C}^n$.

Our conclusions about the benefits to the backward error of scaling $Q$ apply equally well to the condition numbers. Indeed, using (5.14) the analysis in [7] can be improved to provide bounds for $\kappa_L/\kappa_{\widetilde{Q}}$ expressed in terms of $\omega$ instead of $\rho$ for $L = C_1$, $L_1$, and $L_2$. Therefore for these three choices of $L$ both backward error (modulo the potential requirement that $\|\widetilde{A}^{-1}\|$, $\|\widetilde{C}^{-1}\| = O(1)$ for $L_1$ and $L_2$) and conditioning are essentially optimal for the scaled problem if $\omega = O(1)$.

## 6. Numerical experiments.

We illustrate the theory on three symmetric QEPs. Our experiments were performed in MATLAB 7, for which the unit roundoff is $u = 2^{-53} \approx 1.1 \times 10^{-16}$. The eigenpairs of $L(\lambda)$ were computed by MATLAB's function `qz`. Table 6.1 reports the problem sizes, the coefficient matrix norms, and the values of $\rho$ in (3.18) (or (5.9)) before and after scaling via (5.10). In our figures, the $x$-axis is the eigenvalue index and the eigenvalues are sorted in increasing order

Table 6.1
*Problem statistics. Here,* $|\lambda_{\min}| = \min_i |\lambda_i|$, $|\lambda_{\max}| = \max_i |\lambda_i|$.

| Problem | Wave | | Nuclear | | Mass-spring | |
|---|---|---|---|---|---|---|
| $n$ | 25 | | 8 | | 50 | |
| | Unscaled | Scaled | Unscaled | Scaled | Unscaled | Scaled |
| $\lvert\lambda_{\min}\rvert$ | 1.0e0 | 4.0e-2 | 1.8e1 | 6.6e-2 | 1.6e-2 | 7.0e-3 |
| $\lvert\lambda_{\max}\rvert$ | 2.5e1 | 1.0e0 | 3.6e2 | 1.4e0 | 3.2e2 | 1.4e2 |
| $\lVert A\rVert_2$ | 1.6e0 | 1.9e0 | 2.4e8 | 1.2e0 | 1.0e0 | 1.4e-2 |
| $\lVert B\rVert_2$ | 3.2e0 | 1.5e-1 | 4.4e10 | 8.2e-1 | 3.2e2 | 2.0e0 |
| $\lVert C\rVert_2$ | 9.8e2 | 1.9e0 | 1.7e13 | 1.2e0 | 5.0e0 | 1.4e-2 |
| $\lVert A^{-1}\rVert_2$ | 6.4e-1 | 5.4e-1 | 1.8e-1 | 3.7e7 | 1.0e0 | 7.2e1 |
| $\lVert C^{-1}\rVert_2$ | 6.4e-1 | 3.4e2 | 2.1e-4 | 2.9e9 | 1.0e0 | 3.6e2 |
| $\rho$ | 6.2e2 | 1.0e0 | 7.1e4 | 1.0e0 | 3.2e2 | 1.4e2 |
| | $\tau$ = 8.0e-2, $\max\omega$ = 1.1e0 | | $\tau$ = 7.0e-1, $\max\omega$ = 1.6e0 | | $\tau$ = 1.4e2, $\max\omega$ = 7.2e1 | |

of absolute value. Throughout this section "companion" refers to the first companion linearization, $C_1$.

Our first problem comes from applying the Galerkin method to a PDE describing the wave motion of a vibrating string with clamped ends in a spatially inhomogeneous environment [3], [9]. The quadratic $Q$ is elliptic. Table 6.2 displays the smallest and largest ratios $\eta_Q(x,\alpha,\beta)/\eta_L(z,\alpha,b)$ over all computed eigenvalues for several linearizations and for the two ways of recovering the right eigenvector: $x = z_1$ and $x = z_2$. These ratios are compared with the corresponding theoretical upper bounds (5.2), (5.3), and (5.5)–(5.8) (taking the same $(\alpha,\beta)$ as for the smallest/largest backward error ratio). The upper bounds for the scaled companion linearization are smaller than those for the companion linearization, as expected by the theory since $c \gg 1$, and they also are sharper. For $\mathbb{DL}(Q)$ linearizations, the theory suggests using $L_1$ with $x = z_1$, since all the eigenvalues of $Q$ have modulus at least 1. This is reflected in Table 6.2, where the $L_1, z_1$ pairing produces smaller ratios and upper bounds than $L_2, z_1$. For the scaled quadratic $\widetilde{Q}$ in (5.10), we computed the bounds of Theorem 5.1. Since this problem is elliptic, we know from Theorem 5.1 that for the scaled problem, whose eigenvalues lie between 0.04 and 1 in modulus, the scaled and unscaled companion linearizations and the $\mathbb{DL}(\widetilde{Q})$ linearization $L_2$ will have backward errors similar to those for $\widetilde{Q}$ for every eigenvalue with the choice $x = z_2$. This is confirmed by the boldface entries in the last two columns of Table 6.2.

Our second problem is a simplified model of a nuclear power plant, as described in [12], [23]. The largest ratios $\eta_Q(x,\alpha,\beta)/\eta_L(z,\alpha,b)$ and corresponding upper bounds are displayed in Table 6.3. Similar conclusions to those for the wave problem can be drawn for this problem. Since $\rho = 7 \times 10^4$, it is not surprising that some very large ratios are obtained. This example also illustrates the advantage of scaling the companion matrix. This is even more striking in Figure 6.1, where the ratios for all the right and left eigenpairs are displayed. For the companion linearization, these ratios can be up to $10^{10}$ times as large as those for $D_sC_1$. Although the problem is not elliptic, $\|B\|_2 \le \sqrt{\|A\|_2\|C\|_2}$ holds, and so our theory says that scaling will make the scaled and unscaled companion linearizations and the $\mathbb{DL}(Q)$ linearization $L_2$ with $x = z_2$ (since the scaled eigenvalues have modulus at most 1) optimally stable. This prediction is confirmed by the boldface entries in Table 6.3. Notice that for the scaled quadratic $\widetilde{Q}$, the bounds for $L_1$ with $z_2$ and $L_2$ with $z_1$ are very weak, due to the large values of $\|\widetilde{A}^{-1}\|_2$ and $\|\widetilde{C}^{-1}\|_2$ shown in Table 6.1.

Our third problem is a standard damped mass-spring system, as described in [23,

Table 6.2
*Wave problem, $n = 25$.*

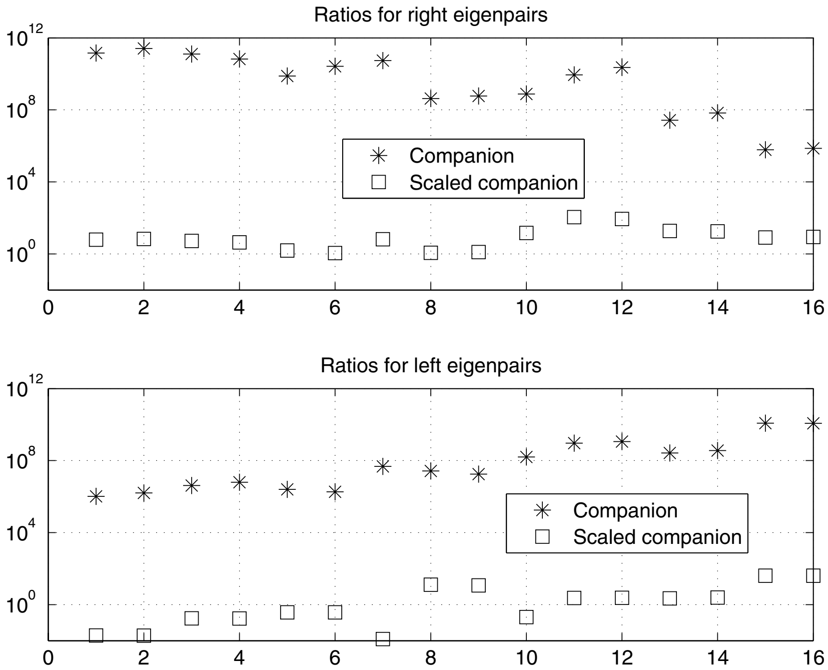
| | | Unscaled, $\rho = 6e2$ | | | | Scaled, $\rho = 1$ | |
|---|---|---|---|---|---|---|---|
| Linearization $L$ | Ei'vec $x$ | $\min \frac{\eta_Q}{\eta_L}$ | Upper bound | $\max \frac{\eta_Q}{\eta_L}$ | Upper bound | $\max \frac{\eta_{\widetilde{Q}}}{\eta_L}$ | Upper bound |
| Companion | $z_1$ | 2.0e1 | 1.7e6 | 3.7e2 | 1.5e6 | 2.9e1 | 3.0e2 |
| | $z_2$ | 1.4e0 | 1.6e4 | 3.8e2 | 3.5e7 | **2.5e0** | **1.6e1** |
| Scaled companion | $z_1$ | 9.8e-1 | 1.6e1 | 1.1e2 | 1.7e3 | 1.4e1 | 1.5e2 |
| | $z_2$ | 6.9e-1 | 1.6e1 | 1.1e2 | 4.3e4 | **1.7e0** | **8.3e0** |
| $L_1$ | $z_1$ | 1.9e0 | 2.2e1 | 9.0e1 | 1.1e3 | 1.8e1 | 7.6e1 |
| | $z_2$ | 1.7e0 | 2.4e1 | 8.3e2 | 6.4e4 | 8.9e1 | 2.8e3 |
| $L_2$ | $z_1$ | 2.2e1 | 9.8e5 | 1.7e2 | 2.0e4 | 1.4e1 | 1.6e2 |
| | $z_2$ | 1.5e0 | 8.0e0 | 1.5e2 | 2.1e4 | **1.7e0** | **8.3e0** |

Table 6.3
*Nuclear problem, $n = 8$.*

| | | Unscaled, $\rho = 7e4$ | | | | Scaled, $\rho = 1$ | |
|---|---|---|---|---|---|---|---|
| Linearization $L$ | Ei'vec $x$ | $\min \frac{\eta_Q}{\eta_L}$ | Upper bound | $\max \frac{\eta_Q}{\eta_L}$ | Upper bound | $\max \frac{\eta_{\widetilde{Q}}}{\eta_L}$ | Upper bound |
| Companion | $z_1$ | 6.1e5 | 3.2e18 | 2.6e11 | 2.8e16 | 4.2e0 | 2.8e2 |
| | $z_2$ | 5.5e5 | 1.2e21 | 9.3e9 | 9.9e19 | **3.1e-1** | **1.8e1** |
| Scaled companion | $z_1$ | 3.4e-1 | 4.1e4 | 2.3e1 | 5.1e4 | 1.0e1 | 1.4e2 |
| | $z_2$ | 3.0e-1 | 4.1e6 | 2.1e1 | 5.9e6 | **2.2e-1** | **9.2e0** |
| $L_1$ | $z_1$ | 2.1e1 | 1.2e3 | 2.3e3 | 1.4e5 | 1.4e1 | 6.9e1 |
| | $z_2$ | 2.0e1 | 1.4e11 | 5.7e5 | 3.2e14 | 8.3e2 | 2.4e10 |
| $L_2$ | $z_1$ | 4.4e3 | 6.0e17 | 4.6e4 | 1.6e17 | 4.4e1 | 3.7e9 |
| | $z_2$ | 1.2e2 | 1.5e4 | 1.9e3 | 6.2e6 | **7.6e-1** | **9.1e0** |

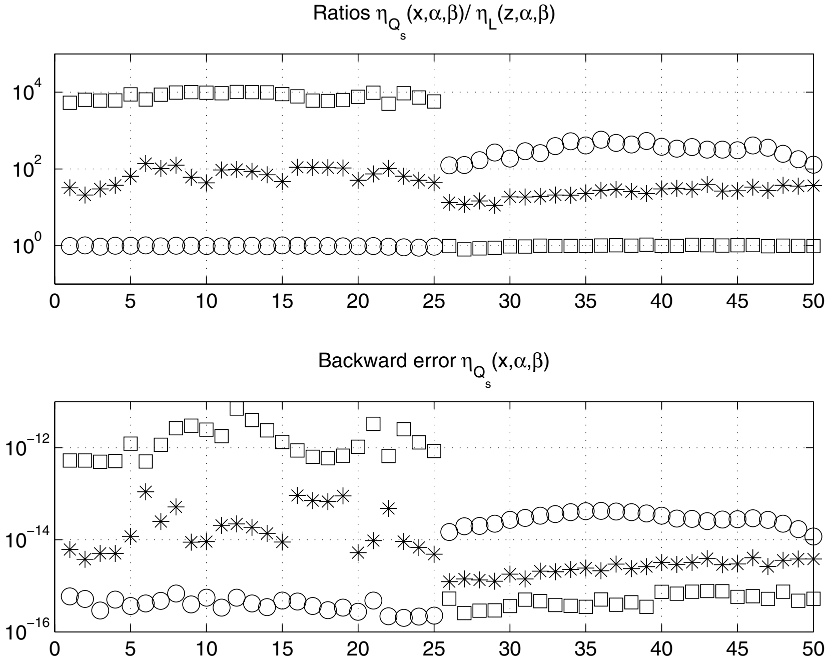
sec. 3.9]. The matrix $A = I$, $B$ is tridiagonal with super- and subdiagonal elements all $-64$ and diagonal $128, 192, 192, \dots, 192$, and $C$ is tridiagonal with super- and subdiagonal elements all $-1$ and diagonal $2, 3, \dots, 3$. The eigenvalues are all negative, with 50 eigenvalues of large modulus ranging from $-320$ to $-6.4$ and 50 small modulus eigenvalues approximately $-1.5\times 10^{-2}$. For the approximate right eigenvector, we take $x = z_1$ if $|\lambda| \geq 1$ and $x = z_2$ otherwise, as suggested by the theory. The largest ratios $\eta_Q(x, \alpha, \beta)/\eta_L(z, \alpha, b)$ and corresponding upper bounds are displayed in Table 6.4. Notice that for this problem the upper bound on the ratio $\eta_Q(x, \alpha, \beta)/\eta_{C_1}(z, \alpha, \beta)$ is nearly attained, which suggests that the factor $\max(1, a, b, c)^2$ in the bound should indeed contain the square. The largest ratio for $L = L_1$ corresponds to a small eigenvalue with $x = z_2$ and, for $L = L_2$, the largest ratio corresponds to a large eigenvalue with $x = z_1$. Hence, the reported upper bounds contain the extra factors $(a + b)\|C^{-1}\|_2$ and $(b + c)\|A^{-1}\|_2$, respectively, which explains why the bounds are larger than those for the scaled companion linearization (on the scaled and unscaled problems), which are small multiples of $\rho$. The top plot in Figure 6.2 shows that for the scaled quadratic $\widetilde{Q}$, small backward error ratios are obtained for $L = L_1$ and large eigenvalues, whereas the ratios are small with the choice $L = L_2$ for the small eigenvalues—all as the theory predicts. The bottom plot in Figure 6.2 confirms that the actual backward errors $\eta_Q$ are what we would expect, given the ratios and the fact that the computed eigenpairs of $L$ are obtained via the QZ algorithm and so necessarily have a backward error of order $u$.

Finally, we mention that further numerical illustration of the bounds developed here, on a symmetric QEP arising from a finite element model of a simply supported beam, can be found in [8].


Ratios for right eigenpairs

Ratios for left eigenpairs

FIG. 6.1. *Nuclear problem. Ratios* $\eta_Q/\eta_L$ *for companion linearization* $L = C_1$ *and scaled companion linearization* $L = D_sC_1$ *for right eigenpairs* (*top*) *and left eigenpairs* (*bottom*).


Ratios $\eta_{Q_s}(x,\alpha,\beta)/\eta_L(z,\alpha,\beta)$

Backward error $\eta_{Q_s}(x,\alpha,\beta)$

FIG. 6.2. *Damped mass-spring problem. Ratios* $\eta_{Q_s}(x,\alpha,\beta)/\eta_L(z,\alpha,\beta)$ *and actual backward errors* $\eta_{Q_s}(x,\alpha,\beta)$ *with* $x = z_1$ *if* $|\alpha| \geq |\beta|$ *and* $x = z_2$ *otherwise, for* $L = D_sC_1$ ($*$) *and for* $L = L_1$ (□) *and* $L = L_2$ (○). *Here,* $Q_s$ *denotes the scaled quadratic* $\widetilde{Q}$.

TABLE 6.4
*Damped mass-spring problem, $n = 50$.*

| | Unscaled, $\rho = 3e2$ | | Scaled, $\rho = 1e2$ | |
|---|---|---|---|---|
| Linearization $L$ | $\max \frac{\eta_Q}{\eta_L}$ | Upper bound | $\max \frac{\eta_{\widetilde{Q}}}{\eta_L}$ | Upper bound |
| $C_1$ | 8.8e3 | 5.8e4 | 1.7e2 | 8.1e2 |
| $D_sC_1$ | 1.0e2 | 9.0e2 | 1.4e2 | 4.1e2 |
| $L_1$ | 2.0e3 | 2.9e4 | 1.0e4 | 1.5e5 |
| $L_2$ | 1.8e3 | 1.1e5 | 5.7e2 | 3.8e4 |

**Acknowledgment.** We thank Steve Mackey for helpful discussions regarding the proof of Theorem 3.2.

## REFERENCES

[1] Z. BAI, J. W. DEMMEL, J. J. DONGARRA, A. RUHE, AND H. A. VAN DER VORST, EDS., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, Software Environ. Tools 11, SIAM, Philadelphia, 2000.

[2] H.-Y. FAN, W.-W. LIN, AND P. VAN DOOREN, *Normwise scaling of second order polynomial matrices*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 252–256.

[3] P. FREITAS, M. GRINFIELD, AND P. KNIGHT, *Stability of finite-dimensional systems with indefinite damping*, Adv. Math. Sci. Appl., 17 (1997), pp. 435–446.

[4] I. GOHBERG, M. A. KAASHOEK, AND P. LANCASTER, *General theory of regular matrix polynomials and band Toeplitz operators*, Integral Equations Operator Theory, 11 (1988), pp. 776–882.

[5] I. GOHBERG, P. LANCASTER, AND L. RODMAN, *Matrix Polynomials*, Academic Press, New York, 1982.

[6] N. J. HIGHAM, D. S. MACKEY, N. MACKEY, AND F. TISSEUR, *Symmetric linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 29 (2006), pp. 143–159.

[7] N. J. HIGHAM, D. S. MACKEY, AND F. TISSEUR, *The conditioning of linearizations of matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1005–1028.

[8] N. J. HIGHAM, D. S. MACKEY, F. TISSEUR, AND S. D. GARVEY, *Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems*, MIMS EPrint 2006.406, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2006. Internat. J. Numer. Methods Engrg., to appear.

[9] N. J. HIGHAM, F. TISSEUR, AND P. M. VAN DOOREN, *Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems*, Linear Algebra Appl., 351–352 (2002), pp. 455–474.

[10] A. HILLIGES, C. MEHL, AND V. MEHRMANN, *On the solution of palindromic eigenvalue problems*, in Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), Jyväskylä, Finland, P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, eds., 2004. Available online from http://www.mit.jyu.fi/eccomas2004/proceedings/proceed.html.

[11] I. C. F. IPSEN, *Accurate eigenvalues for fast trains*, SIAM News, 37 (2004), pp. 1–2.

[12] T. ITOH, *Damped vibration mode superposition method for dynamic response analysis*, Earthquake Engrg. Struct. Dyn., 2 (1973), pp. 47–57.

[13] P. LANCASTER, *Quadratic eigenvalue problems*, Linear Algebra Appl., 150 (1991), pp. 499–506.

[14] P. LANCASTER AND P. PSARRAKOS, *A Note on Weak and Strong Linearizations of Regular Matrix Polynomials*, MIMS EPrint 2006.72, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2006.

[15] P. LANCASTER AND M. TISMENETSKY, *The Theory of Matrices*, 2nd ed., Academic Press, London, 1985.

[16] D. LEMONNIER AND P. M. VAN DOOREN, *Balancing regular matrix pencils*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 253–263.

[17] D. S. MACKEY, *Structured Linearizations for Matrix Polynomials*, Ph.D. thesis, University of Manchester, Manchester, UK, 2006.

[18] D. S. MACKEY, N. MACKEY, C. MEHL, AND V. MEHRMANN, *Structured polynomial eigenvalue problems: Good vibrations from good linearizations*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1029–1051.

[19] D. S. Mackey, N. Mackey, C. Mehl, and V. Mehrmann, *Vector spaces of linearizations for matrix polynomials*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 971–1004.

[20] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM J. Numer. Anal., 10 (1973), pp. 241–256.

[21] G. W. Stewart, *On a companion operator for analytic functions*, Numer. Math., 18 (1971), pp. 26–43.

[22] F. Tisseur, *Backward error and condition of polynomial eigenvalue problems*, Linear Algebra Appl., 309 (2000), pp. 339–361.

[23] F. Tisseur and K. Meerbergen, *The quadratic eigenvalue problem*, SIAM Rev., 43 (2001), pp. 235–286.

[24] P. M. Van Dooren and P. Dewilde, *The eigenstructure of an arbitrary polynomial matrix: Computational aspects*, Linear Algebra Appl., 50 (1983), pp. 545–579.

[25] R. C. Ward, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 141–152.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1242–1246

# FURTHER RESULTS ON THE REVERSE ORDER LAW FOR GENERALIZED INVERSES*

DRAGAN S. DJORDJEVIĆ†

**Abstract.** The reverse order rule $(AB)^\dagger = B^\dagger A^\dagger$ for the Moore–Penrose inverse is established in several equivalent forms. Results related to other generalized inverses are also proved.

**Key words.** Moore–Penrose inverse, generalized inverses, reverse order law

**AMS subject classifications.** 47A05, 15A09

**DOI.** 10.1137/050638114

## 1. Introduction.

Throughout this paper $\mathcal{H}, \mathcal{K}, \mathcal{L}$ denote arbitrary Hilbert spaces. We use $\mathcal{L}(\mathcal{H}, \mathcal{K})$ to denote the set of all linear bounded operators from $\mathcal{H}$ to $\mathcal{K}$. Also, $\mathcal{L}(\mathcal{H}) = \mathcal{L}(\mathcal{H}, \mathcal{H})$. For $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$, we use $\mathcal{R}(A)$ to denote the range, and $\mathcal{N}(A)$ to denote the null-space of $A$. The Moore–Penrose inverse of $A$ is denoted by $A^\dagger$. It is well known that the Moore–Penrose inverse of $A$ exists if and only if $\mathcal{R}(A)$ is closed. We assume that the reader is familiar with the properties of the Moore–Penrose inverse (see, for example, [1], [4], [7], [10], [11], [12]). We also assume that the following classes of operators are well known: $A\{1\}$, $A\{1,3\}$, $A\{1,4\}$, $A\{1,2,3\}$, $A\{1,2,4\}$.

Some equivalent conditions of the reverse order rule

$$(AB)^\dagger = B^\dagger A^\dagger \tag{1}$$

are well known (see all references). We shall prove some new conditions, which are equivalent to (1). Also, conditions

$$\begin{aligned} B\{1,3\} \cdot A\{1,3\} &\subset (BA)\{1,3\}, \\ B\{1,4\} \cdot A\{1,3\} &\subset (BA)\{1,4\}, \\ B^\dagger A^\dagger &\in (AB)\{1,2,3\}, \\ B^\dagger A^\dagger &\in (AB)\{1,2,4\}, \\ B^\dagger A^\dagger &\in (AB)\{1,3\}, \\ B^\dagger A^\dagger &\in (AB)\{1,4\} \end{aligned}$$

will be investigated. By now, some of these conditions are investigated for complex matrices.

The aim of this paper is to prove some equivalence results for linear bounded Hilbert space operators, and thus obtain well-known results connected to the reverse order rule (1).

---

*Received by the editors August 12, 2005; accepted for publication (in revised form) by H. J. Werner May 10, 2007; published electronically December 13, 2007. This work was supported by the Ministry of Science of Serbia under grant 144003.

http://www.siam.org/journals/simax/29-4/63811.html

†Department of Mathematics and Informatics, Faculty of Sciences and Mathematics, University of Niš, P. O. Box 224, 18000 Niš, Serbia (dragan@pmf.ni.ac.yu, ganedj@EUnet.yu).

**2. Results.** We begin with the following auxiliary result, which can be found in [1] for complex matrices. For completeness, we give its proof.

LEMMA 2.1. *Let* $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$ *have a closed range and* $B \in \mathcal{L}(\mathcal{K},\mathcal{H})$. *Then the following statements are equivalent:*

(1) $ABA = A$ *and* $(AB)^* = AB$;

(2) *there exists some* $X \in \mathcal{L}(\mathcal{K},\mathcal{H})$ *such that* $B = A^\dagger + (I - A^\dagger A)X$.

*Proof.* (2) $\Longrightarrow$ (1): Obvious. (1) $\Longrightarrow$ (2): Since $A = \begin{bmatrix} A_1 & 0 \\ 0 & 0 \end{bmatrix} : \begin{bmatrix} \mathcal{R}(A^*) \\ \mathcal{N}(A) \end{bmatrix} \to \begin{bmatrix} \mathcal{R}(A) \\ \mathcal{N}(A^*) \end{bmatrix}$, where $A_1$ is invertible, it follows that $A^\dagger = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}$. An elementary calculation shows that $B = \begin{bmatrix} A_1^{-1} & 0 \\ U & V \end{bmatrix}$, where $U, V$ are arbitrary linear and bounded. Now, take $X = \begin{bmatrix} X_1 & X_2 \\ U & V \end{bmatrix}$ for arbitrary $X_1, X_2$ linear and bounded. ☐

Now, we prove the main result of this paper.

THEOREM 2.2. *Let* $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$ *and* $B \in \mathcal{L}(\mathcal{K},\mathcal{L})$ *be such that* $A, B, AB$ *have closed ranges. Then the following statements are equivalent:*

(1) $\mathcal{R}(A^*AB) \subset \mathcal{R}(B)$;

(2) $B\{1,3\} \cdot A\{1,3\} \subset (AB)\{1,3\}$;

(3) $B^\dagger A^\dagger \in (AB)\{1,3\}$;

(4) $B^\dagger A^\dagger \in (AB)\{1,2,3\}$.

*Proof.* The operator $B$ has the following matrix form with respect to the orthogonal sum of subspaces: $B = \begin{bmatrix} B_1 & 0 \\ 0 & 0 \end{bmatrix} : \begin{bmatrix} \mathcal{R}(B^*) \\ \mathcal{N}(B) \end{bmatrix} \to \begin{bmatrix} \mathcal{R}(B) \\ \mathcal{N}(B^*) \end{bmatrix}$, where $B_1$ is invertible. From the proof of Lemma 2.1 it follows that any $B^{(1,3)} \in B\{1,3\}$ has the form $\begin{bmatrix} B_1^{-1} & 0 \\ U & V \end{bmatrix}$. The operator $A$ has the following form: $A = \begin{bmatrix} A_1 & A_2 \\ 0 & 0 \end{bmatrix} : \begin{bmatrix} \mathcal{R}(B) \\ \mathcal{N}(B^*) \end{bmatrix} \to \begin{bmatrix} \mathcal{R}(A) \\ \mathcal{N}(A^*) \end{bmatrix}$. Now, $A^* = \begin{bmatrix} A_1^* & 0 \\ A_2^* & 0 \end{bmatrix}$ and $AA^* = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$, where $D = A_1A_1^* + A_2A_2^*$ is positive and invertible in $\mathcal{L}(\mathcal{R}(\mathcal{A}))$. We obtain $A^\dagger = A^*(AA^*)^\# = \begin{bmatrix} A_1^*D^{-1} & 0 \\ A_2^*D^{-1} & 0 \end{bmatrix}$. Let $A^{(1,3)} \in A\{1,3\}$. By Lemma 2.1 it follows that there exists some $X \in \mathcal{L}(\mathcal{L},\mathcal{K})$ such that $A^{(1,3)} = A^\dagger + (I - A^\dagger A)X$. Let $X$ have the form $X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} : \begin{bmatrix} \mathcal{R}(A) \\ \mathcal{N}(A^*) \end{bmatrix} \to \begin{bmatrix} \mathcal{R}(B) \\ \mathcal{N}(B^*) \end{bmatrix}$. We then get the following:

$$A^{(1,3)} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

and

$$ABB^{(1,3)}A^{(1,3)} = \begin{bmatrix} A_1Z_{11} & A_1Z_{12} \\ 0 & 0 \end{bmatrix},$$

where

$$\begin{aligned} Z_{11} &= A_1^*D^{-1} + (I - A_1^*D^{-1}A_1)X_{11} - A_1^*D^{-1}A_2X_{21}, \\ Z_{12} &= (I - A_1^*D^{-1}A_1)X_{12} - A_1^*D^{-1}A_2X_{22}, \\ Z_{21} &= A_2^*D^{-1} - A_2^*D^{-1}A_1X_{11} + (I - A_2^*D^{-1}A_2)X_{21}, \\ Z_{22} &= -A_2^*D^{-1}A_1X_{12} + (I - A_2^*D^{-1}A_2)X_{22}. \end{aligned}$$

Notice also that $A^*AB = \begin{bmatrix} A_1^*A_1B_1 & 0 \\ A_2^*A_1B_1 & 0 \end{bmatrix}$.

(1) $\Longrightarrow$ (2): The inclusion $\mathcal{R}(A^*AB) \subset \mathcal{R}(B)$ is equivalent to $BB^\dagger A^*AB = A^*AB$. Now, $BB^\dagger A^*AB = \begin{bmatrix} A_1^*A_1B_1 & 0 \\ 0 & 0 \end{bmatrix}$. Hence, $BB^\dagger A^*AB = A^*AB$ is equivalent to

$A_2^*A_1B_1 = 0$. Since $B_1$ is invertible, we obtain $A_2^*A_1 = 0$, or, equivalently, $A_1^*A_2 = 0$. It follows that $\mathcal{R}(A_2) \subset \mathcal{N}(A_1^*)$. We have the following orthogonal decomposition: $\mathcal{R}(A) = \overline{\mathcal{R}(A_1)} \oplus \mathcal{N}(A_1^*)$. Now,

$$\begin{aligned}\mathcal{R}(A) &= \left\{\left[\begin{smallmatrix} A_1x+A_2y \\ 0 \end{smallmatrix}\right] : x \in \mathcal{R}(B), y \in \mathcal{N}(B^*)\right\} = \mathcal{R}(A_1) + \mathcal{R}(A_2)\\ &= \mathcal{R}(A_1) \oplus \mathcal{R}(A_2),\end{aligned}$$

knowing that $\mathcal{R}(A_2) \subset \mathcal{N}(A_1^*)$. Since $\mathcal{R}(A)$ is closed, we get that both $\mathcal{R}(A_1)$ and $\mathcal{R}(A_2)$ are closed. Consider the following decompositions of $A_1$ and $A_2$: $A_1 = \left[\begin{smallmatrix} A_{11} & 0 \\ 0 & 0 \end{smallmatrix}\right] : \left[\begin{smallmatrix} \mathcal{R}(A_1^*) \\ \mathcal{N}(A_1) \end{smallmatrix}\right] \to \left[\begin{smallmatrix} \mathcal{R}(A_1) \\ \mathcal{N}(A_1^*) \end{smallmatrix}\right]$, where $A_{11}$ is invertible, and $A_2 = \left[\begin{smallmatrix} 0 & 0 \\ A_{22} & 0 \end{smallmatrix}\right] : \left[\begin{smallmatrix} \mathcal{R}(A_2^*) \\ \mathcal{N}(A_2) \end{smallmatrix}\right] \to \left[\begin{smallmatrix} \mathcal{R}(A_1) \\ \mathcal{N}(A_1^*) \end{smallmatrix}\right]$. We have the following: $0 < D = A_1A_1^* + A_2A_2^* = \left[\begin{smallmatrix} A_{11}A_{11}^* & 0 \\ 0 & A_{22}A_{22}^* \end{smallmatrix}\right]$, implying that both $A_{11}A_{11}^*$ and $A_{22}A_{22}^*$ are invertible. Hence, $D^{-1} = \left[\begin{smallmatrix} (A_{11}A_{11}^*)^{-1} & 0 \\ 0 & (A_{22}A_{22}^*)^{-1} \end{smallmatrix}\right]$. Notice that $A_1^*D^{-1}A_1 = \left[\begin{smallmatrix} I & 0 \\ 0 & 0 \end{smallmatrix}\right]$, $A_1(I - A_1^*D^{-1}A_2) = 0$, and $A_1^*D^{-1}A_2 = 0$. Now, it follows that

$$A_1[(I - A_1^*D^{-1}A_1)X_{12} - A_1^*D^{-1}A_2X_{22}] = 0$$

and

$$A_1[A_1^*D^{-1} + (I - A_1^*D^{-1}A_1)X_{11} - A_1^*D^{-1}A_2X_{21}] = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$$

is selfadjoint. An elementary computation shows that $ABB^{(1,3)}A^{(1,3)}AB = AB$.

(2) $\Longrightarrow$ (3): Obvious.

(3) $\Longrightarrow$ (1): From the proof of the implication (1) $\Longrightarrow$ (2), it follows that the condition $\mathcal{R}(A^*AB) \subset \mathcal{R}(B)$ is equivalent to $A_2^*A_1 = 0$. Now, $ABB^\dagger A^\dagger = \left[\begin{smallmatrix} A_1A_1^*D^{-1} & 0 \\ 0 & 0 \end{smallmatrix}\right]$ is selfadjoint, implying that $[A_1A_1^*, D^{-1}] = 0 = [A_1A_1^*, D]$ (here $[U, V] = UV - VU$). Also, $\left[\begin{smallmatrix} A_1B_1 & 0 \\ 0 & 0 \end{smallmatrix}\right] = AB = ABB^\dagger A^\dagger AB = \left[\begin{smallmatrix} A_1A_1^*D^{-1}B_1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, implying that $A_1B_1 = A_1A_1^*D^{-1}A_1B_1 = D^{-1}A_1A_1^*A_1B_1$. Hence, we get $DA_1B_1 = A_1A_1^*A_1B_1$ and, consequently, $A_2A_2^*A_1B_1 = 0$. Since $B_1$ is invertible, we obtain $A_2A_2^*A_1 = 0$ and $\mathcal{R}(A_1) \subset \mathcal{N}(A_2A_2^*) = \mathcal{N}(A_2^*)$. It follows that $A_2^*A_1 = 0$.

(4) $\Longrightarrow$ (3): Obvious.

(1) $\Longrightarrow$ (4): If $\mathcal{R}(A^*AB) \subset \mathcal{R}(B)$, we have to prove that $B^\dagger A^\dagger ABB^\dagger A^\dagger = B^\dagger A^\dagger$. Notice that $AB = \left[\begin{smallmatrix} A_1B_1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$ and $B^\dagger A^\dagger = \left[\begin{smallmatrix} B_1^{-1}A_1^*D^{-1} & 0 \\ 0 & 0 \end{smallmatrix}\right]$. By using previously proved facts, $D$ commutes with $A_1A_1^*$ (the implication (3) $\Longrightarrow$ (1)) and matrix forms of $A_1$ and $D$ (the implication (1) $\Longrightarrow$ (2)), which we compute as follows:

$$\begin{aligned}B_1^{-1}A_1^*D^{-1}A_1B_1B_1^{-1}A_1^*D^{-1} &= B_1^{-1}A_1^*A_1A_1^*D^{-2}\\ &= B_1^{-1}A_1^*\begin{bmatrix} A_{11}A_{11}^* & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} (A_{11}A_{11}^*)^{-2} & 0 \\ 0 & (A_{22}A_{22}^*)^{-2} \end{bmatrix} \qquad (2)\\ &= B_1^{-1}A_1^*\begin{bmatrix} (A_{11}A_{11}^*)^{-1} & 0 \\ 0 & 0 \end{bmatrix} = B_1^{-1}A_1^*D^{-1}.\end{aligned}$$

Now, it obviously follows that $B^\dagger A^\dagger ABB^\dagger A^\dagger = B^\dagger A^\dagger$ is satisfied. □

In the same manner we can prove the following result.

THEOREM 2.3. *Let $A \in \mathcal{L}(\mathcal{H}, \mathcal{K})$ and $B \in \mathcal{L}(\mathcal{K}, \mathcal{L})$ be such that $A, B, AB$ have closed ranges. Then the following statements are equivalent:*

(1) *$\mathcal{R}(BB^*A^*) \subset \mathcal{R}(A^*)$;*
(2) *$B\{1,4\} \cdot A\{1,4\} \subset (AB)\{1,4\}$;*
(3) *$B^\dagger A^\dagger \in (AB)\{1,4\}$;*
(4) *$B^\dagger A^\dagger \in (AB)\{1,2,4\}$.*

For complex matrices, see the following literature: The equivalence (1) $\Longleftrightarrow$ (4) in both Theorems 2.2 and 2.3 is proved in [15]; conditions (2) in both Theorems 2.2 and 2.3 are investigated in [16].

Now, as a corollary, we obtain the following result.

COROLLARY 2.4. *Let $A \in \mathcal{L}(\mathcal{H},\mathcal{K})$ and $B \in \mathcal{L}(\mathcal{K},\mathcal{L})$ be such that $A, B, AB$ have closed ranges. Then the following statements are equivalent:*

(1) *$\mathcal{R}(A^*AB) \subset \mathcal{R}(B)$ and $\mathcal{R}(BB^*A^*) \subset \mathcal{R}(A^*)$;*
(2) *$B\{1,3\} \cdot A\{1,3\} \subset AB\{1,3\}$ and $B\{1,4\} \cdot A\{1,4\} \subset AB\{1,4\}$;*
(3) *$B^\dagger A^\dagger \in AB\{1,3,4\}$;*
(4) *$B^\dagger A^\dagger = (AB)^\dagger$.*

It is important to mention that the equivalence (1) $\Longleftrightarrow$ (4) is a classical result, proved for complex matrices in [6], and for bounded operators on Hilbert spaces in [2], [3], and [9].

*Remark* 1. The equivalence (3) $\Longleftrightarrow$ (4) in Theorems 2.2, 2.3 and Corollary 2.4 suggests that the "{2}-property" is implied by the rest. For matrices, this follows from a rank argument. If $X$ is a $\{1\}$-inverse of $A$, then $X$ is also a $\{2\}$-inverse if and only if $\operatorname{rank} X = \operatorname{rank} A$. Since we cannot talk about "rank" here, we resolve this situation using the special partition of operators.

Results which are related to the reverse order rule for generalized inverses follow. Multiple matrix products are considered in [8] and [14]. General conditions to the reverse order rule for inner inverses are given in [18] and for outer inverses in [5]. The reverse order rule for the weighted Moore–Penrose inverse is investigated in [13].

Finally, we find that results of this paper are closely connected with the results of Werner [17]. Although in [17] the finite dimensional technique is used, the results which will be presented here are valid in arbitrary Hilbert spaces also.

In [17] the geometric approach is involved, taking the range and the null-space of the generalized inverses. Among other things, the following result is proved in [17, Theorem 5.5] (interpreted in an infinite dimensional setting).

THEOREM 2.5. *Let $B \in \mathcal{L}(\mathcal{H},\mathcal{K})$ and $A \in \mathcal{L}(\mathcal{K},\mathcal{L})$, such that $A$, $B$, and $C = AB$ have closed ranges. Let $T$ be a closed subspace of $H$ such that $T \overset{\bullet}{+} \mathcal{N}(B) = H$ (the sum is not necessarily orthogonal) and $\mathcal{R}(C^*) \subset T$. Then the following statements are equivalent:*

(1) *There exist some operators $A^-$ and $B^-$ satisfying $AA^-A = A$, $A^-A = P_{\mathcal{R}(A^*),\mathcal{N}(A)}$, $BB^-B = B$, $B^-B = P_{T,\mathcal{N}(B)}$, $BB^- = P_{\mathcal{R}(B),\mathcal{N}(B^*)}$ such that the following is satisfied: $D = B^-A^-$, $CDC = C$, and $DC = P_{\mathcal{R}(C^*),\mathcal{N}(C)}$.*
(2) *$\mathcal{R}(BB^*A) \subset \mathcal{R}(A^*)$.*
(3) *For each operator $A^-$ and $B^-$ satisfying $AA^-A = A$, $A^-A = P_{\mathcal{R}(A^*),\mathcal{N}(A)}$, $BB^-B = B$, $B^-B = P_{T,\mathcal{N}(B)}$, $BB^- = P_{\mathcal{R}(B),\mathcal{N}(B^*)}$, the following holds: $D = B^-A^-$, $CDC = C$, and $DC = P_{\mathcal{R}(C^*),\mathcal{N}(C)}$.*

We see that for $C = AB$ the condition $\mathcal{R}(C^*) \subset \mathcal{R}(B^*)$ holds. Hence, for $T = \mathcal{R}(B^*)$ we get the result closely related to our Theorem 2.3. Now, the corollary is stated according to our notations.

COROLLARY 2.6. *Let $B \in \mathcal{L}(\mathcal{H},\mathcal{K})$ and $A \in \mathcal{L}(\mathcal{K},\mathcal{L})$, such that $A$, $B$, and $C = AB$ have closed ranges. Then the following statements are equivalent:*

(1) *There exist some* $A^- \in A\{1,4\}$ *and some* $B^- \in B\{1,3,4\}$ *such that* $B^-A^- \in C\{1,4\}$.
(2) $\mathcal{R}(BB^*A^*) \subset \mathcal{R}(A^*)$.
(3) $A\{1,4\} \cdot B\{1,3,4\} \subset C\{1,4\}$.

We see that Corollary 2.6 contains a weaker result than our Theorem 2.3.

**Acknowledgments.** I am grateful to Professor Hans Joachim Werner for sending me his paper [17], and for remarks that his work in [17] is closely related to the present results. I am also grateful to the referee for helpfully pointing to Remark 1.

## REFERENCES

[1] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., Springer-Verlag, New York, 2003.

[2] R. H. Bouldin, *The pseudo-inverse of a product*, SIAM J. Appl. Math., 24 (1973), pp. 489–495.

[3] R. H. Bouldin, *Generalized inverses and factorizations*, in Recent Applications of Generalized Inverses, Res. Notes in Math. 66, Pitman, Boston, 1982, pp. 233–249.

[4] S. R. Caradus, *Generalized Inverses and Operator Theory*, Queen's Papers in Pure and Applied Mathematics 50, Queen's University, Kingston, Ontario, Canada, 1978.

[5] D. S. Djordjević, *Unified approach to the reverse order rule for generalized inverses*, Acta Sci. Math. (Szeged), 67 (2001), pp. 761–776.

[6] T. N. E. Greville, *Note on the generalized inverse of a matrix product*, SIAM Rev., 8 (1966), pp. 518–521.

[7] R. E. Harte, *Invertibility and Singularity for Bounded Linear Operators*, Marcel Dekker, New York, 1988.

[8] R. E. Hartwig, *The reverse order law revisited*, Linear Algebra Appl., 76 (1986), pp. 241–246.

[9] S. Izumino, *The product of operators with closed range and an extension of the reverse order law*, Tohoku Math J. (2), 34 (1982), pp. 43–52.

[10] J. J. Koliha, *The Drazin and Moore-Penrose inverse in $C^*$-algebras*, Math. Proc. R. Ir. Acad., 99A (1999), pp. 17–27.

[11] M. Z. Nashed, *Inner, outer, and generalized inverses in Banach and Hilbert spaces*, Numer. Funct. Anal. Optim., 9 (1987), pp. 261–325.

[12] M. Z. Nashed and G. F. Votruba, *A unified operator theory of generalized inverses*, in Generalized Inverses and Applications, M. Z. Nashed, ed., Academic Press, New York, 1976, pp. 1–109.

[13] W. Sun and Y. Wei, *Inverse order rule for weighted generalized inverse*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 772–775.

[14] Y. Tian, *Reverse order laws for the generalized inverses of multiple matrix products*, Generalized Inverses, Linear Algebra Appl., 211 (1994), pp. 85–100.

[15] Y. Tian, *Using rank formulas to characterize equalities for Moore-Penrose inverses of matrix products*, Appl. Math. Comput., 147 (2004), pp. 581–600.

[16] M. Wei and W. Guo, *Reverse order laws for least squares g-inverses and minimum norm g-inverses of products of two matrices*, Linear Algebra Appl., 342 (2002), pp. 117–132.

[17] H. J. Werner, *G-inverses of matrix products*, in Data Analysis and Statistical Inference, S. Schach and G. Trenkler, eds., Bergisch Gladbach, Germany, 1992, pp. 531–546.

[18] H. J. Werner, *When is $B^-A^-$ a generalized inverse of $AB$?*, Linear Algebra Appl., 210 (1994), pp. 255–263.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1247–1266

# A SUPERFAST ALGORITHM FOR TOEPLITZ SYSTEMS OF LINEAR EQUATIONS*

S. CHANDRASEKARAN[†], M. GU[‡], X. SUN[§], J. XIA[¶], AND J. ZHU[‡]

**Abstract.** In this paper we develop a new superfast solver for Toeplitz systems of linear equations. To solve Toeplitz systems many people use displacement equation methods. With displacement structures, Toeplitz matrices can be transformed into Cauchy-like matrices using the FFT or other trigonometric transformations. These Cauchy-like matrices have a special property, that is, their off-diagonal blocks have small numerical ranks. This low-rank property plays a central role in our superfast Toeplitz solver. It enables us to quickly approximate the Cauchy-like matrices by structured matrices called *sequentially semiseparable* (SSS) matrices. The major work of the constructions of these SSS forms can be done in precomputations (independent of the Toeplitz matrix entries). These SSS representations are compact because of the low-rank property. The SSS Cauchy-like systems can be solved in linear time with linear storage. Excluding precomputations the main operations are the FFT and SSS system solve, which are both very efficient. Our new Toeplitz solver is stable in practice. Numerical examples are presented to illustrate the efficiency and the practical stability.

**Key words.** displacement equation, SSS structure, superfast algorithm, Toeplitz matrix

**AMS subject classifications.** 15A06, 65F05, 65G05

**DOI.** 10.1137/040617200

## 1. Introduction.

Toeplitz systems of linear equations arise in many applications, including PDE solving, signal processing, time series analysis, orthogonal polynomials, and many others. A *Toeplitz system* is a linear system

$$Tx = b \tag{1.1}$$

with a coefficient matrix to be a *Toeplitz matrix*

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & t_{-(N-1)} \\ t_1 & t_0 & t_{-1} & \cdots & t_{-(N-2)} \\ t_2 & t_1 & t_0 & \cdots & t_{-(N-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{N-1} & t_{N-2} & t_{N-3} & \cdots & \cdots \end{pmatrix}, \tag{1.2}$$

that is, its entries are constant along every diagonal (a matrix whose entries are constant along every antidiagonal is called a Hankel matrix). The vector $t = (t_{-(N-1)} \ \cdots \ t_{-1} \ t_0 \ t_1 \ \cdots \ t_{N-1})$ is called the *Toeplitz-vector* that generates $T$.

There are both direct and iterative methods for solving (1.1). Direct solvers are said to be *fast* if they cost $O(N^2)$ operations; examples include Schur-type methods,

---

[*] Received by the editors October 19, 2004; accepted for publication (in revised form) by D. A. Bini May 8, 2007; published electronically December 13, 2007.
http://www.siam.org/journals/simax/29-4/61720.html

[†] Department of Electrical and Computer Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106 (shiv@ece.ucsb.edu).

[‡] Department of Mathematics, University of California at Berkeley, Berkeley, CA 94720 (mgu@math.berkeley.edu, zhujiang.cal@gmail.com).

[§] Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 (sunxt@cal.berkeley.edu).

[¶] Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 (jxia@math.ucla.edu).

Levinson-type methods, and others [32]. An important type of direct solver is the *displacement equation–type* fast solver based on Gaussian eliminations. Some known displacement equation–type methods are the Heinig [28], GKO [22], and Gu [26] methods. Those methods have complexity $O(N^2)$. Methods with complexity less than $O(N^2)$ are called *superfast*. In this paper we will present a displacement equation–type superfast algorithm.

**1.1. Fast and superfast methods.** Many fast and superfast methods that have been developed are numerically unstable [5, 15, 16, 7, 37]. References [5] and [15] showed that the Schur algorithm and the Levinson algorithm are weakly stable in some cases, but both may be highly unstable in the case of an indefinite and nonsymmetric matrix. Stable generalized Schur algorithms [32] and look-ahead algorithms were developed in [9, 10]. High-performance look-ahead Schur algorithms were presented [20].

Many other solvers use the FFT or other trigonometric transforms to convert the Toeplitz (or even Hankel or Toeplitz-plus-Hankel) matrices into generalized Cauchy or Vandermonde matrices, which can be done stably in $O(N \log N)$ operations. This is also the approach that we will use in this paper, with the aid of a displacement structure.

The concept of displacement structure was first introduced in [31]. The Sylvester-type *displacement equation* for a matrix $\hat{C} \in \mathbf{R}^{N\times N}$ [29] is

$$\Omega\hat{C} - \hat{C}\Lambda = UV, \tag{1.3}$$

where $\Omega,\ \Lambda \in \mathbf{R}^{N\times N}$, $U \in \mathbf{R}^{N\times\alpha}$, $V \in \mathbf{R}^{\alpha\times N}$, and $\alpha \le N$ is the *displacement rank* with respect to $\Omega$ and $\Lambda$ if $\operatorname{rank}(UV) = \alpha$. The matrix $\hat{C}$ is considered to possess a *displacement structure* with respect to $\Omega$ and $\Lambda$ if $\alpha \ll N$.

With displacement structures it was shown in [19, 24, 36, 22, 28] that Toeplitz and Hankel matrices can be transformed into *Cauchy-like* matrices of the following form:

$$\hat{C} = \left(\frac{u_i^T \cdot v_j}{\eta_i - \lambda_j}\right)_{1\le i,j\le N} \qquad (u_i, v_j \in \mathbf{R}^{\alpha})\ ,$$

where we assume that $\eta_i \ne \lambda_j$ for $1 \le i, j \le N$. Equivalently, a Cauchy-like matrix is the unique solution to the displacement equation (1.3) with

$$\Omega = \operatorname{diag}(\eta_1, \ldots, \eta_n),\ \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n),\ U = \begin{pmatrix} u_1^T \\ \vdots \\ u_n^T \end{pmatrix},\ \text{and } V = (v_1, \ldots, v_n)\,.$$

In particular, $\hat{C}$ is a Cauchy matrix if $u_i^T v_j = 1$ for all $i$ and $j$. The displacement rank of $\hat{C}$ is at most $\alpha$.

To solve Toeplitz systems through Cauchy-like matrices, many people have utilized matrix factorizations. Gohberg and Olshevsky [23] presented a fast variation of the straightforward Gaussian elimination with partial pivoting (GEPP) procedure to solve a Cauchy-like linear system of equations in $O(N^2)$ operations. Among other results, Gohberg, Kailath, and Olshevsky [22] developed algorithm GKO, an improved version of Heinig's algorithm [28], and demonstrated numerically that it is stable. In their algorithm, the Hankel matrix and the Toeplitz-plus-Hankel matrix are also transformed via fast trigonometric transforms into Cauchy-like matrices.

Gu presented a modified algorithm in [26] to avoid extra error growth. This algorithm is numerically stable, provided that the element growth in the computed factorization is not large. The algorithm takes $O(N^2)$ operations and is a fast stable method.

Superfast algorithms appeared in [34, 4, 6, 17, 35, 1, 2, 21] and many others. Superfast algorithms use divide-and-conquer strategies. Morf developed the first idea in [34]. These methods are unstable for nonsymmetric systems as they cannot deal with nearly singular leading principal submatrices.

Van Barel and Kravanja presented a superfast method for rational interpolation at roots of unity [39]. A similar idea was then applied to Toeplitz systems [38]. It provided an explicit formula for the inverse of a Toeplitz matrix. Additional techniques such as iterative refinement and downdating were still required to stabilize their algorithm.

### 1.2. Main results.

Our new Toeplitz solver is also of the displacement equation type. Given a Toeplitz linear system, we first use the FFT to transform the associated Toeplitz matrix into a Cauchy-like matrix. Then instead of using matrix factorizations which often cost $O(N^2)$ or more, we exploit a special *low-rank property* of Cauchy-like matrices, that is, every off-diagonal block of a Cauchy-like matrix has a low numerical rank. Using this low-rank property, we then approximate the Cauchy-like matrix by a low-rank matrix structure called the *sequentially semiseparable* (SSS) matrix proposed by Chandrasekaran et al. [12, 13]. A system with the coefficient matrix in *compact* SSS form can be solved with only $O(p^2N)$ operations, where $N$ is the matrix dimension, and $p$ is the complexity of the semiseparable description. The SSS solver is practically stable in our numerical tests and those in [12, 13].

The SSS structure was developed to capture the low-rank property of the off-diagonal blocks of a matrix and to maintain stability or practical stability in the mean time. It is a matrix analog of semiseparable integral kernels in Kailath's paper [30]. Matrix operations with compact form SSS representations are very efficient, provided that such compact representations exist or can be easily computed. This turns out to be true for our case, as the Cauchy-like matrices are transformed from Toeplitz matrices. We use a recursive compression scheme with a shifting strategy to construct compact SSS forms for those Cauchy-like matrices. The major work of the compressions can be precomputed on some Cauchy matrices which are independent of the actual Toeplitz matrix entries.

The overall algorithm thus has the following stages:

(1) Precompute compressions of off-diagonal blocks of Cauchy matrices.
(2) Transform the Toeplitz matrix into a Cauchy-like matrix in $O(N \log N)$ operations.
(3) Construct a compact SSS representation from precomputed compressions and solve the Cauchy-like matrix system $O(p^2N)$ operations.
(4) Recover the solution of the Toepliz system in $O(N \log N)$ operations.

The stages above are either stable or practically stable. Our numerical results indicate that the overall algorithm is stable in practice. The Toeplitz matrix does not have to be symmetric or positive definite, and no extra stabilizing step is necessary. After the precomputations, the total cost for the algorithm is $O(N \log N) + O(p^2N)$. This indicates that the entire algorithm is superfast.

We also point out that similar techniques are used in [33], where the low-rank property is exploited through the block columns without diagonals (called *neutered block columns* in [33]), in contrast with the off-diagonal blocks here. The compressions of either the neutered blocks or the off-diagonal blocks both give data-sparse representations which enable fast factorizations of the Cauchy-like matrices. In fact, corresponding to neutered block rows or columns, there are also matrix representations called hierarchically semiseparable (HSS) matrices [14] which are usually more complicated structures than SSS matrices.

**1.3. Overview.** We will discuss the displacement structure and the transformation from a Toeplitz problem to a Cauchy-like problem in section 2. The low-rank property of this Cauchy-like problem is then exploited. Section 3 then gives a linear complexity solver using the SSS structure. In section 4, we will present an algorithm for fast construction of SSS structures. We will then analyze the complexity in section 5 and use some numerical experiments to demonstrate the efficiency and the practical stability. All algorithms have been implemented in Fortran 90. Section 6 draws some conclusions.

## 2. Displacement structures and low-rank property.

**2.1. Cauchy-like systems.** Given a Toeplitz system (1.1), we can use a displacement structure to transform it into a Cauchy-like system. Define

$$Z_\delta = \begin{pmatrix} 0 & 0 & \cdots & 0 & \delta \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix},$$

and let $\Omega = Z_1$ and $\Lambda = Z_{-1}$ in (1.3). Kailath, Kung, and Morf [31] have shown that every Toeplitz matrix satisfies the displacement equation (1.3) with $A \cdot B$, having nonzero entries only in its first row and last column, to be a matrix of rank at most 2. Hence the displacement rank of a Toeplitz matrix is at most 2 with respect to $Z_1$ and $Z_{-1}$. The following result can be found in [28].

Proposition 2.1. *Let $\hat{C} \in \mathbf{R}^{N\times N}$ be a matrix satisfying the displacement equation*

$$Z_1\hat{C} - \hat{C}Z_{-1} = UV, \tag{2.1}$$

*where $U \in \mathbf{R}^{n\times\alpha}$ and $V \in \mathbf{R}^{\alpha\times n}$. Then $\mathcal{F}\hat{C}D_0^{-1}\mathcal{F}^H$ is a Cauchy-like matrix satisfying*

$$\mathcal{D}_1(\mathcal{F}\hat{C}D_0^{-1}\mathcal{F}^H) - (\mathcal{F}\hat{C}D_0^{-1}\mathcal{F}^H)\mathcal{D}_{-1} = (\mathcal{F}U)\left(VD_0^H\mathcal{F}^H\right), \tag{2.2}$$

*where $\mathcal{F} = \sqrt{\frac{1}{N}}(\omega^{2(k-1)(j-1)})_{1\le k,j\le N}$ is the normalized inverse discrete Fourier transform matrix, $\omega = e^{\frac{\pi i}{N}}$, and*

$$\mathcal{D}_1 = \mathrm{diag}(1, \omega^2, \ldots, \omega^{2(N-1)}), \quad \mathcal{D}_{-1} = \mathrm{diag}(\omega, \omega^3, \ldots, \omega^{2N-1}),$$
$$D_0 = \mathrm{diag}(1, \omega, \ldots, \omega^{N-1}).$$

Here $\alpha \le 2$. This proposition suggests that for a Toeplitz matrix $T$, one can convert it into the Cauchy-like matrix in (2.2). Therefore the Toeplitz system (1.1) can be readily transformed into a new system

$$C\tilde{x} = \tilde{b}, \tag{2.3}$$

where $C$ has the form

$$C = \left( \frac{u_i^T v_j}{\omega^{2i} - \omega^{2j+1}} \right)_{1 \le i,j \le N} \qquad (u_i,\ v_j \in \mathbf{R}^{\alpha}) \ . \tag{2.4}$$

In section 3 we will present a fast solver for (2.3). After obtaining $\tilde{x}$ we will then recover $x$ with an FFT again. All the stages involving FFT are stable and cost $O(N \log N)$. The solver for (2.3) has a linear complexity and turns out to be practically stable. Thus the total cost of our algorithm is bounded by $O(N \log N) + O(Np^2)$, where $p$ is some parameter that will be described below. This indicates our method is a superfast one with practical stability.

**2.2. Low-rank property of Cauchy-like matrices.** In this section, we will show a low-rank property of $C$, i.e., every off-diagonal block of $C$ has a low numerical rank. This property is the basis of the superfast SSS solver in section 3.

First, a simple numerical experiment can give us an idea of the low-rank property of $C$. To find out the numerical ranks we can use one of the following tools:

1. *$\tau$-accurate SVD*: singular values less than $\tau$ are dropped if $\tau$ is an absolute tolerance, or singular values less than $\tau$ times the largest singular value are dropped if $\tau$ is a relative tolerance.
2. *$\tau$-accurate QR*:

$$A \approx QR,\ A : m \times n,\ \ Q : m \times k,\ \ R : k \times k,\ \ k \le l \equiv \min(m, n),$$

which is obtained in the following way. Compute the exact QR factorization of matrix $A = \hat{Q}\hat{R}$, where $\hat{Q}$ is $m \times l$ and $\hat{R}$ is $l \times n$ with diagonal entries satisfying $\hat{R}_{11} \ge \hat{R}_{22} \ge \cdots \ge \hat{R}_{ll}$. Then obtain $R$ by dropping all rows of $\hat{R}$ with diagonal entries less than $\tau$ if $\tau$ is an absolute tolerance, or with diagonal entries less than $\tau\hat{R}_{11}$ if $\tau$ is a relative tolerance. Drop relevant columns of $\hat{Q}$ accordingly to obtain $Q$.

Later, by ranks we mean numerical ranks. Here we take some random Toeplitz matrices in different sizes. Then we transform them into Cauchy-like matrices $C$ and compute the numerical ranks of their off-diagonal blocks. For simplicity, we compute the ranks for blocks $C(1 : d, d+1 : N)$, $C(1 : 2d, 2d+1 : N)$, $\ldots$, $C(1 : kd, kd+1 : N), \ldots$, where $d$ is a fixed integer, and these blocks are numbered as block numbers $1, 2, \ldots, k$ as shown in Figure 2.1.

Here, $k = 8$ off-diagonal blocks for each of three $N \times N$ Cauchy-like matrices are considered. See Table 2.1 for the numerical ranks, where we use the $\tau$-accurate SVD with $\tau$ to be an absolute tolerance.

We can see that the numerical ranks are relatively small as compared to the block sizes. And when we double the dimension of the matrix, the numerical ranks do not increase much. This is more significant when a larger $\tau$ is used.

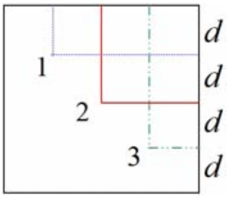


FIG. 2.1. *Off-diagonal blocks (numbered as 1, 2, 3). Upper triangular part only.*

TABLE 2.1
*Off-diagonal numerical ranks of Cauchy-like matrices transformed from some Toeplitz matrices with absolute tolerance* $\tau = 10^{-9}$.

| | $N = 640$ | | $N = 1280$ | | $N = 2560$ | |
|---|---|---|---|---|---|---|
| Block # | Block size | Rank | Block size | Rank | Block size | Rank |
| 1 | $80 \times 560$ | 37 | $160 \times 1120$ | 44 | $320 \times 2240$ | 52 |
| 2 | $160 \times 480$ | 43 | $320 \times 960$ | 50 | $640 \times 1920$ | 57 |
| 3 | $240 \times 400$ | 45 | $480 \times 800$ | 53 | $960 \times 1600$ | 60 |
| 4 | $320 \times 320$ | 46 | $640 \times 640$ | 53 | $1280 \times 1280$ | 61 |
| 5 | $400 \times 240$ | 46 | $800 \times 480$ | 52 | $1600 \times 960$ | 60 |
| 6 | $480 \times 160$ | 43 | $960 \times 320$ | 50 | $1920 \times 640$ | 58 |
| 7 | $560 \times 80$ | 37 | $1120 \times 160$ | 44 | $2240 \times 320$ | 52 |

The low-rank property can be verified theoretically in the following way. We first consider a special case of (2.4) where all $u_i^T v_j = 1$. We show that the following Cauchy matrix has low-rank off-diagonal blocks:

$$C_0 = \left( \frac{1}{\omega^{2i} - \omega^{2j+1}} \right)_{1 \le i,j \le N} \tag{2.5}$$

The central idea is similar to that in the fast multipole method [25, 8], which implies that with *well-separated* points (see, e.g., Figure 2.2) an interaction matrix is numerically low-rank.

Here we introduce two sets of points $\{\lambda_k\}_{k=1}^N \equiv \{\omega^{2k}\}_{k=1}^N$ and $\{\eta_k\}_{k=1}^N \equiv \{\omega^{2k+1}\}_{k=1}^N$ on the unit circle. When we consider an off-diagonal block of $C_0$ as follows:

$$G = \left( \frac{1}{\lambda_i - \eta_j} \right)_{1 \le i \le p,\ p+1 \le j \le N}$$

we can show $G$ is (numerically) low-rank. In fact $G$ corresponds to two well-separated sets $\{\lambda_k\}_{k=1}^p$ and $\{\eta_j\}_{k=p+1}^N$; that is, there exists a point $c \in \mathbb{C}$ such that

$$\begin{array}{rcll} |\lambda_i - c| & > & d + e, & i = 1, \ldots, p, \\ |\eta_j - c| & < & d, & j = p+1, \ldots, N, \end{array} \tag{2.6}$$

where $d, e$ are positive constants. Consider the expansion

$$\frac{1}{\lambda_i - \eta_j} = \frac{1}{\lambda_i - c} \frac{1}{1 - \frac{\eta_j - c}{\lambda_i - c}} = \sum_{k=0}^{r} \frac{(\eta_j - c)^k}{(\lambda_i - c)^{k+1}} + O\left( \left( \frac{\eta_j - c}{\lambda_i - c} \right)^{r+1} \right) \tag{2.7}$$

$$= \sum_{k=0}^{r} \frac{(\eta_j - c)^k}{(\lambda_i - c)^{k+1}} + \varepsilon, \tag{2.8}$$

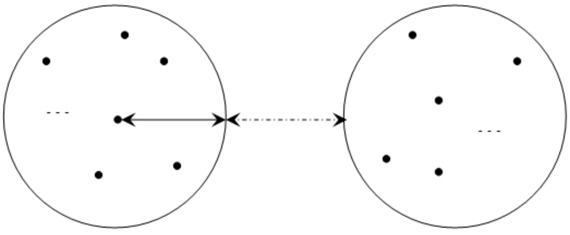


FIG. 2.2. *Well-separated sets in the plane.*

where $r$ is a number such that the error term $|\varepsilon| = |O((\frac{\eta_j - c}{\lambda_i - c})^{r+1})|$ is bounded by a given tolerance. We have the estimate

$$\left|\frac{\eta_j - c}{\lambda_i - c}\right|^{r+1} < \left(\frac{d}{d+e}\right)^{r+1},$$

which enables us to find an appropriate $r$ according to the tolerance. Thus

$$G = \left(\sum_{k=0}^{r} \frac{(\eta_j - c)^k}{(\lambda_i - c)^{k+1}}\right)_{1\le i\le p,\ p+1\le j\le N} + \hat{\varepsilon}$$

(2.9)

$$= \begin{pmatrix} \frac{1}{\lambda_1 - c} & \frac{1}{(\lambda_1 - c)^2} & \cdots & \frac{1}{(\lambda_1 - c)^{r+1}} \\ \frac{1}{\lambda_2 - c} & \frac{1}{(\lambda_2 - c)^2} & \cdots & \frac{1}{(\lambda_2 - c)^{r+1}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{\lambda_p - c} & \frac{1}{(\lambda_p - c)^2} & \cdots & \frac{1}{(\lambda_p - c)^{r+1}} \end{pmatrix} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ (\eta_{p+1} - c) & (\eta_{p+2} - c) & \cdots & (\eta_N - c) \\ \vdots & \vdots & \cdots & \vdots \\ (\eta_{p+1} - c)^r & (\eta_{p+2} - c)^r & \cdots & (\eta_N - c)^r \end{pmatrix} + \hat{\varepsilon}.$$

Therefore the numerical rank of $G$ is at most $r + 1$, up to an error $\hat{\varepsilon}$.

Now we can return to the Cauchy-like matrix (2.4). A similar argument shows that any off-diagonal block of $C$ satisfies

$$\hat{G} \approx \left(\sum_{k=0}^{r} (u_i^T v_j) \frac{(\eta_j - c)^k}{(\lambda_i - c)^{k+1}}\right)_{1\le i\le p,\ p+1\le j\le N}$$

$$= \begin{pmatrix} \frac{u_1^T}{\lambda_1 - c} & \frac{u_1^T}{(\lambda_1 - c)^2} & \cdots & \frac{u_1^T}{(\lambda_1 - c)^{r+1}} \\ \frac{u_2^T}{\lambda_2 - c} & \frac{u_2^T}{(\lambda_2 - c)^2} & \cdots & \frac{u_2^T}{(\lambda_2 - c)^{r+1}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{u_p^T}{\lambda_p - c} & \frac{u_p^T}{(\lambda_p - c)^2} & \cdots & \frac{u_p^T}{(\lambda_p - c)^{r+1}} \end{pmatrix} \begin{pmatrix} v_1 & v_2 & \cdots & v_q \\ (\eta_{p+1} - c)\, v_1 & (\eta_{p+2} - c)\, v_2 & \cdots & (\eta_N - c)\, v_N \\ \vdots & \vdots & \cdots & \vdots \\ (\eta_{p+1} - c)^r v_1 & (\eta_{p+2} - c)^r v_2 & \cdots & (\eta_N - c)^r v_N \end{pmatrix}.$$

That is, we replace the entries of the two matrix factors in (2.9) with appropriate vectors. Thus the numerical rank of $\hat{G}$ will be no larger than $\alpha(r + 1)$, which will be relatively small as compared to $N$.

## 3. SSS structures and superfast SSS solver.

### 3.1. SSS representations.

To take advantage of the low-rank property of the Cauchy-like matrix $C$, we can use SSS structures introduced by Chandrasekaran et al. [12, 13]. The SSS structure nicely captures the ranks of off-diagonal blocks of a matrix such as shown in Figure 2.1.

A matrix $A \in \mathbf{C}^{M\times\tilde{M}}$ satisfies the SSS structure if there exist $2n$ positive integers $m_1, \ldots, m_n$, and $\tilde{m}_1, \ldots, \tilde{m}_n$ with $M = m_1 + \cdots + m_n$ and $\tilde{M} = \tilde{m}_1 + \cdots + \tilde{m}_n$ to block-partition $A$ as $A = (A_{i,j})_{k\times k}$, where $A_{ij} \in \mathbb{C}^{m_i\times\tilde{m}_j}$ satisfies

$$A_{ij} = \begin{cases} D_i & \text{if } i = j, \\ U_i W_{i+1} \cdots W_{j-1} V_j^H & \text{if } i < j, \\ P_i R_{i-1} \cdots R_{j+1} Q_j^H & \text{if } i > j. \end{cases} \tag{3.1}$$

Here the superscript $H$ denotes the Hermitian transpose and empty products are defined to be identity matrices. The matrices $\{U_i\}_{i=1}^{n-1}$, $\{V_i\}_{i=2}^{n}$, $\{W_i\}_{i=2}^{n-1}$, $\{P_i\}_{i=2}^{n}$,

Table 3.1
*Dimensions of matrices in* (3.1).

| Matrix | $U_i$ | $V_i$ | $W_i$ | $P_i$ | $Q_i$ | $R_i$ |
|---|---|---|---|---|---|---|
| Dimensions | $m_i \times k_i$ | $\tilde{m}_i \times k_{i-1}$ | $k_{i-1} \times k_i$ | $m_i \times l_i$ | $\tilde{m}_i \times l_{i+1}$ | $l_{i+1} \times l_i$ |

$\{Q_i\}_{i=1}^{n-1}$, $\{R_i\}_{i=2}^{n-1}$, and $\{D_i\}_{i=1}^{n}$ are called *generators* for the SSS structure and their dimensions are defined in Table 3.1.

As an example, the matrix $A$ with $n = 4$ has the form

$$A = \begin{pmatrix} D_1 & U_1V_2^H & U_1W_2V_3^H & U_1W_2W_3V_4^H \\ P_2Q_1^H & D_2 & U_2V_3^H & U_2W_3V_4^H \\ P_3R_2Q_1^H & P_3Q_2^H & D_3 & U_3V_4^H \\ P_4R_3R_2Q_1^H & P_4R_3Q_2^H & P_4Q_3^H & D_4 \end{pmatrix}. \tag{3.2}$$

The SSS representation (3.2) is related to the off-diagonal blocks in Figure 2.1 in the way that the upper off-diagonal block numbers $1, 2,$ and $3$ are

$$U_1 \begin{pmatrix} V_2^H & W_2V_3^H & W_2W_3V_4^H \end{pmatrix}, \begin{pmatrix} U_1W_2 \\ U_2 \end{pmatrix} \begin{pmatrix} V_3^H & W_3V_4^H \end{pmatrix}, \begin{pmatrix} U_1W_2W_3 \\ U_2W_3 \\ U_3 \end{pmatrix} V_4^H.$$

Appropriate row and column bases of the off-diagonal blocks are clearly reflected.

The SSS structure depends on the sequences $\{m_i\}$ and $\{\tilde{m}_i\}$ and the SSS generation scheme. If $A$ is a square matrix ($M = \tilde{M}$), then we can have a simpler situation $m_i = \tilde{m}_i$, $i = 1, \ldots, n$. SSS matrices are closed under addition, multiplication, inversion, etc., although the sizes of the generators may increase.

While any matrix can be represented in this form for sufficiently large $k_i$'s and $l_i$'s, the column dimensions of $U_i$'s and $P_i$'s, respectively, our main focus will be on SSS matrices which have low-rank off-diagonal blocks and have generators with $k_i$'s and $l_i$'s to be close to those ranks. We say these SSS matrices are *compact*. Particularly true for Cauchy-like matrices, they can have compact SSS forms. Using SSS structures, we can take advantage of the superfast SSS system solver in [12, 13] to solve the Cauchy-like systems. The solver is efficient when the SSS form is compact, and is practically stable. The solver shares similar ideas with that for banded plus semiseparable systems in [11].

Here we briefly describe the main ideas of the solver in [12, 13]. We consider solving the linear system $Ax = b$, where $A \in \mathbf{C}^{N\times N}$ satisfies (3.1) and $b$ itself is an unstructured matrix. The solver computes an implicit $ULV^H$ decomposition of $A$, where $U$ and $V$ are orthogonal matrices.

Before we present the formal algorithm, we demonstrate the key ideas on a $4 \times 4$ block matrix example.

**3.2. SSS solver: $4 \times 4$ example.** Let the initial system $Ax = b$ be partitioned as follows:

(3.3)

$$\begin{pmatrix} D_1 & U_1V_2^H & U_1W_2V_3^H & U_1W_2W_3V_4^H \\ P_2Q_1^H & D_2 & U_2V_3^H & U_2W_3V_4^H \\ P_3R_2Q_1^H & P_3Q_2^H & D_3 & U_3V_4^H \\ P_4R_3R_2Q_1^H & P_4R_3Q_2^H & P_4Q_3^H & D_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} - \begin{pmatrix} 0 \\ P_2 \\ P_3R_2 \\ P_4R_3R_2 \end{pmatrix} \xi,$$

where the dimensions of the generators follow those in Table 3.1 with $m_i = \tilde{m}_i$ and the vector $\xi = 0$. The extra zero vector $\xi$ on the right-hand side of (3.3) has been added for the purpose of a general recursive pattern.

The algorithm has two main stages, compression (or elimination) and merging, depending on the relationship between $k_i$ ($l_i$) and $m_i$ in the intermediate procedure.

**3.2.1. Compression.** At the beginning, $k_1 < m_1$ because of the low-rank property described earlier. We apply a unitary transformation $q_1^H$ to $U_1$ so that the first $m_1 - k_1$ rows of $U_1$ become zeros:

$$q_1^H U_1 = \begin{array}{cc} \begin{pmatrix} 0 \\ \hat{U}_1 \end{pmatrix} & \begin{array}{c} m_1 - k_1 \\ k_1 \end{array} \end{array}. \tag{3.4}$$

Now we multiply $q_1^H$ to the first $m_1$ equations of the system

$$\begin{pmatrix} q_1^H & 0 \\ 0 & I \end{pmatrix} Ax = \begin{pmatrix} q_1^H & 0 \\ 0 & I \end{pmatrix} b - \begin{pmatrix} 0 \\ P_2 \\ P_3R_2 \\ P_4R_3R_2 \end{pmatrix} \xi.$$

We pick another unitary transformation $w_1^H$ to lower-triangularize $q_1^H D_1$, the $(1,1)$ diagonal block $A$, i.e.,

$$\left(q_1^H D_1\right) w_1^H = \begin{array}{c} \\ m_1 - k_1 \\ k_1 \end{array} \begin{array}{c} \begin{array}{cc} m_1 - k_1 & k_1 \end{array} \\ \begin{pmatrix} D_{11} & 0 \\ D_{21} & D_{22} \end{pmatrix} \end{array}.$$

Then system (3.3) becomes

$$\begin{pmatrix} q_1^H & 0 \\ 0 & I \end{pmatrix} A \begin{pmatrix} w_1^H & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} w_1 & 0 \\ 0 & I \end{pmatrix} x = \begin{pmatrix} q_1^H & 0 \\ 0 & I \end{pmatrix} b - \begin{pmatrix} 0 \\ P_2 \\ P_3R_2 \\ P_4R_3R_2 \end{pmatrix} \xi,$$

which can be rewritten as

$$\begin{pmatrix} D_{11} & 0 & 0 & 0 & 0 \\ D_{21} & D_{22} & \hat{U}_1V_2^H & \hat{U}_1W_2V_3^H & \hat{U}_1W_2W_3V_4^H \\ P_2Q_{11}^H & P_2\hat{Q}_1^H & D_2 & U_2V_3^H & U_2W_3V_4^H \\ P_3R_2Q_{11}^H & P_3R_2\hat{Q}_1^H & P_3Q_2^H & D_3 & U_3V_4^H \\ P_4R_3R_2Q_{11}^H & P_4R_3R_2\hat{Q}_1^H & P_4R_3Q_2^H & P_4Q_3^H & D_4 \end{pmatrix} \begin{pmatrix} z_1 \\ \hat{x}_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$= \begin{pmatrix} \beta_1 \\ \gamma_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ P_2 \\ P_3R_2 \\ P_4R_3R_2 \end{pmatrix} \xi,$$

where we have used the partitions

$$w_1x_1 = \begin{array}{c} m_1 - k_1 \\ k_1 \end{array} \begin{pmatrix} z_1 \\ \hat{x}_1 \end{pmatrix}, \quad q_1^H b_1 = \begin{array}{c} m_1 - k_1 \\ k_1 \end{array} \begin{pmatrix} \beta_1 \\ \gamma_1 \end{pmatrix}, \quad \text{and} \quad w_1Q_1 = \begin{array}{c} m_1 - k_1 \\ k_1 \end{array} \begin{pmatrix} Q_{11} \\ \hat{Q}_1 \end{pmatrix}.$$

At this point, we can solve for $z_1$ from the system of equations $D_{11}z_1 = \beta_1$. We also subtract $D_{21}z_1$ from the right-hand side to obtain $\hat{b}_1 = \gamma_1 - D_{21}z_1$. Then we can discard the first $m_1 - k_1$ rows and columns of the coefficient matrix of the system to obtain

$$\begin{pmatrix} D_{22} & \hat{U}_1 V_2^H & \hat{U}_1 W_2 V_3^H & \hat{U}_1 W_2 W_3 V_4^H \\ P_2 \hat{Q}_1^H & D_2 & U_2 V_3^H & U_2 W_3 V_4^H \\ P_3 R_2 \hat{Q}_1^H & P_3 Q_2^H & D_3 & U_3 V_4^H \\ P_4 R_3 R_2 \hat{Q}_1^H & P_4 R_3 Q_2^H & P_4 Q_3^H & D_4 \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} - \begin{pmatrix} 0 \\ P_2 \\ P_3 R_2 \\ P_4 R_3 R_2 \end{pmatrix} \hat{\xi},$$

where $\hat{\xi} = \xi + Q_{11}^H z_1$. This new system has a similar structure to the original one but with smaller dimension. We can continue to solve it by recursion, if further compressions of the blocks such as (3.4) are possible. Note the actual solution, say, $x_1$, can be recovered by

$$x_1 = w_1^H \begin{pmatrix} z_1 \\ \hat{x}_1 \end{pmatrix}.$$

**3.2.2. Merging.** During the recursive eliminations there are situations when $k_i$ is no longer smaller than $m_i$ and no further compression is possible. We are then unable to introduce more zeros into the system. Now we proceed by merging appropriate block rows and columns of the matrix. As an example we can merge the first two block rows and columns and rewrite the system of equations as follows:

$$\begin{pmatrix} \begin{pmatrix} D_1 & U_1 V_2^H \\ P_2 Q_1^H & D_2 \end{pmatrix} & \begin{pmatrix} U_1 W_2 \\ U_2 \end{pmatrix} V_3^H & \begin{pmatrix} U_1 W_2 \\ U_2 \end{pmatrix} W_3 V_4^H \\ P_3 \begin{pmatrix} Q_1 R_2^H \\ Q_2 \end{pmatrix}^H & D_3 & U_3 V_4^H \\ P_4 R_3 \begin{pmatrix} Q_1 R_2^H \\ Q_2 \end{pmatrix}^H & P_4 Q_3^H & D_4 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ x_3 \\ x_4 \end{pmatrix}$$

$$= \begin{pmatrix} \begin{pmatrix} b_1 \\ b_2 - P_2 \hat{\xi} \end{pmatrix} \\ b_3 \\ b_4 \end{pmatrix} - \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ P_3 \\ P_4 R_3 \end{pmatrix} (R_2 \hat{\xi}).$$

Hence the system becomes

$$\begin{pmatrix} \hat{D}_1 & \hat{U}_1 V_3^H & \hat{U}_1 W_3 V_4^H \\ P_3 \hat{Q}_1^H & D_3 & U_3 V_4^H \\ P_4 R_3 \hat{Q}_1^H & P_4 Q_3^H & D_4 \end{pmatrix} \begin{pmatrix} \hat{x}_1 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ b_3 \\ b_4 \end{pmatrix} - \begin{pmatrix} 0 \\ P_3 \\ P_4 R_3 \end{pmatrix} (\tilde{\xi}),$$

where

$$\hat{D}_1 = \begin{pmatrix} D_1 & U_1 V_2^H \\ P_2 Q_1^H & D_2 \end{pmatrix}, \quad \hat{U}_1 = \begin{pmatrix} U_1 W_2 \\ U_2 \end{pmatrix}, \quad \hat{Q}_1 = \begin{pmatrix} Q_1 R_2^H \\ Q_2 \end{pmatrix},$$

$$\hat{x}_1 = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \hat{b}_1 = \begin{pmatrix} b_1 \\ b_2 - P_2 \tau \end{pmatrix}, \quad \tilde{\xi} = R_2 \hat{\xi}.$$

The number of block rows/columns is reduced by one. Further compressions become possible and we can proceed to solve the system recursively. In the case $n = 1$, we have the system $D_1 x_1 = b_1 - 0\xi$, which is solved directly.

**3.3. General solve algorithm.** We now present a short description of the general algorithm. The procedure in the $4 \times 4$ example can be directly extended to a general system. We assume that the matrix $A$ is in compact SSS form represented by the generators $\{U_i\}_{i=1}^{n-1}$, $\{V_i\}_{i=2}^{n}$, $\{W_i\}_{i=2}^{n-1}$, $\{P_i\}_{i=2}^{n}$, $\{Q_i\}_{i=1}^{n-1}$, $\{R_i\}_{i=2}^{n-1}$, and $\{D_i\}_{i=1}^{n}$ as in (3.1). We also partition $x = (x_i)$ and $b = (b_j)$ such that $x_i$ and $b_i$ have $m_i$ rows. As in the $4 \times 4$ example, there are two stages at each step of the recursion.

In the compression stage, we perform orthogonal eliminations on both sides of $A$ to create an $(m_1 - k_1) \times (m_1 - k_1)$ lower triangular submatrix at the top left corner of $A$. Then we solve a small triangular system and obtain the first few components of the solution vector. At this stage, we are left with a new system with less unknowns; hence we can carry out a recursion.

In the merging stage, we merge the first two block rows and columns of $A$ while still maintaining the SSS structure. The numbers of block rows and columns are reduced by one.

Combining these two stages, we can proceed with recursion to solve the system. When $n = 1$ we can solve the linear system directly with standard solvers.

The SSS solver has a complexity $O(Np^2)$ [12, 13], where $p$ is the maximum numerical rank of the off-diagonal blocks of $A$, as compared to the traditional $O(N^3)$ cost for a general dense $N \times N$ matrix. We use only orthogonal transformations and a single substitution in the SSS solver. Although a formal proof for the backward stability is not yet available, the solver is shown to be practically stable. The reader is referred to [12, 13] for more discussions on the stability.

**4. Fast construction of SSS representation for $C$.** According to section 3, a system in compact SSS form can be solved very efficiently. We can thus use that algorithm to solve the Cauchy-like system (2.3), provided that $C$ can be quickly written in SSS form. Therefore we try to find an efficient construction scheme. Here we provide a divide-and-conquer SSS construction scheme by using the fast merging and splitting strategy in [13]. If we know further that the matrix has low-rank off-diagonal blocks and it is easy to compress the off-diagonal blocks, then the construction can be superfast. Here we will concentrate on this situation as Cauchy-like matrices have this low-rank property.

We first present a general divide-and-conquer construction algorithm and then describe a fast shifting strategy to compress the off-diagonal blocks of $C$.

**4.1. General divide-and-conquer construction algorithm.** In this section, we discuss the construction of the SSS structure of a matrix $A$, when the partition sequence $\{m_i\}_{i=1}^{n}$ is given. The general construction methods can be applied to any unstructured matrix, thus proving that any matrix has an SSS structure. (Of course, $k_i$ and $l_i$ will usually be large in this case, precluding any speed-ups.) These methods can be viewed as specific ways to make the realization algorithm of [18] more efficient.

Suppose we are given an $N \times N$ matrix $A$ and a partition sequence $\{m_i\}_{i=1}^{n}$ with $\sum_{i=1}^{n} m_i = N$. Starting with an appropriate sequence $\{\tilde{m}_1, \tilde{m}_2\}$, where $\sum_{i=1}^{k} m_i = \tilde{m}_1$ and $\sum_{i=k+1}^{n} m_i = \tilde{m}_2$, we can first partition $A$ into a $2 \times 2$ block matrix and then construct a simple SSS form

$$A = \begin{array}{c c} & \begin{array}{cc} \tilde{m}_1 & \tilde{m}_2 \end{array} \\ \begin{array}{c} \tilde{m}_1 \\ \tilde{m}_2 \end{array} & \begin{pmatrix} D_1 & B \\ E & D_2 \end{pmatrix} \end{array} = \begin{array}{c} \begin{array}{cc} \tilde{m}_1 & \tilde{m}_2 \end{array} \\ \begin{pmatrix} D_1 & U_1 V_2^H \\ P_2 Q_1^H & D_2 \end{pmatrix} \end{array}, \tag{4.1}$$

where

$$B = U_1 V_2^H \equiv U_1 \left(\Sigma_1 F_1^H\right), \; F = P_2 Q_1^H \equiv P_2 \left(\Sigma_1 F_1^H\right) \tag{4.2}$$

are the low-rank SVDs of the off-diagonal blocks $B$ and $E$. Note if the compressions are done by $\tau$-accurate SVD approximations or rank-revealing QR decompositions, then appropriate "=" signs should be replaced by "≈" signs. We now split either the $(1,1)$ block or the $(2,2)$ block to obtain a $3\times 3$ block SSS matrix. For instance, we can split the $(1,1)$ block according to an appropriate new sequence $\{\hat{m}_1, \hat{m}_2, \hat{m}_3\}$ as follows, where $\hat{m}_1 + \hat{m}_2 = m_1,\ \hat{m}_3 = m_2$:

$$\begin{pmatrix} D_1^{\mathbf{new}} & U_1^{\mathbf{new}}(V_2^{\mathbf{new}})^H \\ P_2^{\mathbf{new}}(Q_1^{\mathbf{new}})^H & D_2^{\mathbf{new}} \end{pmatrix} = D_1,\ \begin{pmatrix} U_1^{\mathbf{new}} W_2^{\mathbf{new}} \\ U_2^{\mathbf{new}} \end{pmatrix} = U_1,\ (V_3^{\mathbf{new}})^H = V_2^H,$$

$$(R_2^{\mathbf{new}}(Q_1^{\mathbf{new}})^H (Q_2^{\mathbf{new}})^H) = Q_1^H,\ P_3^{\mathbf{new}} = P_2,\ D_3^{\mathbf{new}} = D_2,$$

where the new generators (marked by the superscript **new**) introduced based on (4.1) can be determined in the following way. First, we partition the matrices for the old first block conformally with the two new blocks as

$$\begin{pmatrix} D_1^{11} & D_1^{12} \\ D_1^{21} & D_1^{22} \end{pmatrix} = D_1,\ \begin{pmatrix} U_1^1 \\ U_1^2 \end{pmatrix} = U_1,\ ((Q_1^1)^H (Q_1^2)^H) = Q_1^H.$$

We can then identify from these and the previous equations that

$D_1^{\mathbf{new}} = D_1^{11}$, $D_2^{\mathbf{new}} = D_1^{22}$, $U_2^{\mathbf{new}} = U_1^2$, $V_3^{\mathbf{new}} = V_2$, $Q_2^{\mathbf{new}} = Q_1^2$, $P_3^{\mathbf{new}} = P_2$, $D_3^{\mathbf{new}} = D_2$.

The remaining matrices satisfy

$$(D_1^{12} U_1^1) = U_1^{\mathbf{new}}((V_2^{\mathbf{new}})^H W_2^{\mathbf{new}}),\ \begin{pmatrix} D_1^{21} \\ (Q_1^1)^H \end{pmatrix} = \begin{pmatrix} P_2^{\mathbf{new}} \\ R_2^{\mathbf{new}} \end{pmatrix} (Q_1^{\mathbf{new}})^H. \tag{4.3}$$

By factorizing the left-hand side matrices using numerical tools such as the SVD and rank-revealing QR, these two equations allow us to compute those remaining matrices for the new blocks.

$A$ is thus in the new form

$$A = \begin{array}{c} \\ \hat{m}_1 \\ \hat{m}_2 \\ \hat{m}_3 \end{array} \begin{array}{c} \begin{array}{ccc} \hat{m}_1 & \hat{m}_2 & \hat{m}_3 \end{array} \\ \begin{pmatrix} D_1^{\mathbf{new}} & U_1^{\mathbf{new}}(V_2^{\mathbf{new}})^H & U_1^{\mathbf{new}} W_2^{\mathbf{new}} (V_3^{\mathbf{new}})^H \\ P_2^{\mathbf{new}}(Q_1^{\mathbf{new}})^H & D_2^{\mathbf{new}} & U_2^{\mathbf{new}}(V_3^{\mathbf{new}})^H \\ P_2^{\mathbf{new}} R_2^{\mathbf{new}} (Q_1^{\mathbf{new}})^H & P_3^{\mathbf{new}}(Q_2^{\mathbf{new}})^H & D_3^{\mathbf{new}} \end{pmatrix} \end{array}.$$

We can use similar techniques if we want to split the second row and column of (4.1).

We can continue this by either splitting the first block row and block column or the last ones using the above techniques, or splitting any middle block row and block column similarly. Then we will be able to construct the desired SSS representation according to the given sequence $\{m_i,\ i = 1, 2, \ldots, n\}$.

The general construction can be organized with bisection. The major cost is in compressions of off-diagonal blocks of the form

$$D_i^{12} = X_i Y_i^H, \tag{4.4}$$

where $X_i$ and $Y_i$ are tall and thin, and $D_i^{12}$ is an off-diagonal block of

$$D_i = \begin{pmatrix} D_i^{11} & D_i^{12} \\ D_i^{21} & D_i^{22} \end{pmatrix}.$$

The compression (4.4) can be achieved by a $\tau$-accurate QR factorization.

The construction is also practically stable in our implementation.

**4.2. Compression of off-diagonal blocks.** For a general matrix with low-rank off-diagonal blocks, the SSS construction can cost $O(N^2p)$ as a compression such as (4.2), (4.3), and (4.4) can take $O(K^2p)$, where $K$ is the dimension of the block being compressed. However, for the Cauchy-like matrix $C$ in (2.3) the compressions can be precomputed.

THEOREM 4.1. *The compression of the off-diagonal block*

$$G = \left( \frac{u_i^T \cdot v_j}{\lambda_i - \eta_j} \right)_{1 \le i \le p,\ q \le j \le N} = XY^H \tag{4.5}$$

*of* $C$ *can be obtained by the compression of the corresponding off-diagonal block*

$$G = \left( \frac{1}{\lambda_i - \eta_j} \right)_{1 \le i \le p,\ q \le j \le N} = X_0 Y_0^H$$

*of* $C_0$*, where the column dimension of* $X$ *is no larger than twice the size of the column dimension of* $X_0$*.*

*Proof.* Assume the off-diagonal block $G_0 = (\frac{1}{\lambda_i - \eta_j})_{1 \le i \le p,\ q \le j \le N}$ is in compressed form $X_0 Y_0^H$, and the $(i,j)$ entry is $\frac{1}{\lambda_i - \eta_j} = X_{i,:} Y_{j,:}^H$, where $X_{i,:}$ $(Y_{j,:})$ denotes the $i$th row of $X$ $(Y)$. As $\alpha \le 2$ for simplicity we fix $\alpha = 2$ in (2.1). Then the corresponding off-diagonal block in $C$ is

$$\begin{aligned}
G &= \left( \frac{u_i^T \cdot v_j}{\lambda_i - \eta_j} \right)_{1 \le i \le p,\ q \le j \le N} = \left( \frac{u_{i1} v_{j1} + u_{i2} v_{j2}}{\lambda_i - \eta_j} \right)_{1 \le i \le p,\ q \le j \le N} \\
&= \left( (u_{i1} v_{j1} + u_{i2} v_{j2})\, X_{i,:} Y_{j,:}^H \right)_{1 \le i \le p,\ q \le j \le N} \\
&= \left( \begin{pmatrix} u_{i1} X_{i,:} & u_{i2} X_{i,:} \end{pmatrix} \begin{pmatrix} v_{j1} Y_{j,:}^H \\ v_{j2} Y_{j,:}^H \end{pmatrix} \right)_{1 \le i \le p,\ q \le j \le N} \\
&= \begin{pmatrix} u_{11} X_{1,:} & u_{12} X_{1,:} \\ \vdots & \vdots \\ u_{p1} X_{i,:} & u_{p2} X_{i,:} \end{pmatrix} \begin{pmatrix} v_{q1} Y_{q,:}^H & \cdots & v_{N1} Y_{N,:}^H \\ v_{q2} Y_{q,:}^H & \cdots & v_{N2} Y_{N,:}^H \end{pmatrix} \\
&\equiv XY^H.
\end{aligned}$$

That is, we get a compression of $G$. □

Theorem 4.1 indicates that we can convert the compressions of the off-diagonal blocks of $C$ to be the compressions of those of $C_0$ which is independent of the actual entries of the Toeplitz matrix $T$. This means the compressions of off-diagonal blocks of $C_0$ can be precomputed. The precomputation can be done in $O(N^2p)$ flops by a rank-revealing QR factorization such as in [27]. It is possible to reduce the cost to $O(N \log N)$ due to the fact that the compression of a large off-diagonal block can be obtained by that of small ones. This can be seen implicitly from the following subsection.

**4.3. Compressions of off-diagonal blocks in precomputation.** We further present a shifting strategy to reduce the cost of the compressions in the precomputation. The significance of this shifting strategy is to relate the compressions of large off-diagonal block of $C_0$ to those of small ones. That is, in different splitting stages of the SSS construction of $C$, the compressions of off-diagonal blocks with different sizes can be related.

For simplicity, we look at the example of partitioning $C_0$ into $4\times 4$ blocks. Assume in the first cut that we can partition $C_0$ into $2\times 2$ blocks with equal dimensions as in the following:

$$C_0 = \left(\frac{1}{\omega^{2i}-\omega^{2j+1}}\right)_{1\le i,\ j\le N} = \left(\begin{array}{c||c} C_{0;1,1} & C_{0;1,2} \\ \hline\hline C_{0;2,1} & C_{0;2,2} \end{array}\right)$$

$$= \left(\begin{array}{ccc||ccc} \frac{1}{\omega^2-\omega^3} & \cdots & \frac{1}{\omega^2-\omega^{4k+1}} & \frac{1}{\omega^2-\omega^{4k+3}} & \cdots & \frac{1}{\omega^2-\omega^{2N+1}} \\ \vdots & & \vdots & \vdots & \cdots & \vdots \\ \frac{1}{\omega^{4k}-\omega^3} & \cdots & \frac{1}{\omega^{4k}-\omega^{4k+1}} & \frac{1}{\omega^{4k}-\omega^{4k+3}} & \cdots & \frac{1}{\omega^{4k}-\omega^{2N+1}} \\ \hline\hline & & & & & \\ & C_{0;2,1} & & & C_{0;2,2} & \\ & & & & & \end{array}\right),$$

where $k=\frac{N}{4}$, and without loss of generality, we consider only the block upper triangular part. Assume that we have obtained a compression $X_1Y_1^H$ of $C_{0;1,2}$ such that

$$C_0 = \left(\begin{array}{c||c} C_{0;1,1} & X_1Y_1^H \\ \hline\hline C_{0;2,1} & C_{0;2,2} \end{array}\right),$$

where $X_1$ and $Y_1$ are tall and thin and can be computed with $\tau$-accurate SVD or rank-revealing QR factorization. Next, we split $C_{0;1,1}$, the $(1,1)$ block of $C_0$, into $2\times 2$ blocks and compress its off-diagonal block. Suppose the resulting off-diagonal block of $C_{0;1,1}$ has size $k\times(\frac{N}{2}-l)$. We will compress it by shifting certain parts of $X_1$ and $Y_1$. That is, we partition $X_1$ and $Y_1$ conformally as

$$X_1 = \begin{pmatrix} X_{1,1} \\ X_{1,2} \end{pmatrix},\ Y_1 = \begin{pmatrix} Y_{1,1} \\ Y_{1,2} \end{pmatrix},$$

and also pick a $k\times(\frac{N}{2}-l)$ block from the lower left corner of $C_{0;1,2}=X_1Y_1^H$ (that is, $X_{1,2}Y_{1,1}^H$)

$$C_0 = \left(\begin{array}{c||c} C_{0;1,1} & X_1Y_1^H \\ \hline\hline C_{0;2,1} & C_{0;2,2} \end{array}\right)$$

$$= \left(\begin{array}{c|ccc||ccc|c} & \frac{1}{\omega^2-\omega^{2l+3}} & \cdots & \frac{1}{\omega^2-\omega^{N+1}} & & & & \\ & \vdots & \cdots & \vdots & & & & \\ & \frac{1}{\omega^{2k}-\omega^{2l+3}} & \cdots & \frac{1}{\omega^{2k}-\omega^{N+1}} & & & & \\ \hline & & & & \frac{1}{\omega^{2k+2}-\omega^{N+3}} & \cdots & \frac{1}{\omega^{2k+2}-\omega^{2(N-l)+1}} & \\ & & & & \vdots & \cdots & \vdots & \\ & & & & \frac{1}{\omega^{4k}-\omega^{N+3}} & \cdots & \frac{1}{\omega^{N}-\omega^{2(N-l)+1}} & \\ \hline\hline \multicolumn{4}{c||}{} & \multicolumn{4}{c}{} \\ \multicolumn{4}{c||}{C_{0;2,1}} & \multicolumn{4}{c}{C_{0;2,2}} \\ \multicolumn{4}{c||}{} & \multicolumn{4}{c}{} \end{array}\right)$$

$$= \left(\begin{array}{c|c||c|c} & X_2Y_2^H & X_{1,1}Y_{1,1}^H & X_{1,1}Y_{1,2}^H \\ \hline & & X_{1,2}Y_{1,1}^H & X_{1,2}Y_{1,2}^H \\ \hline\hline \multicolumn{2}{c||}{C_{0;2,1}} & \multicolumn{2}{c}{C_{0;2,2}} \end{array}\right),$$

where $X_2Y_2^H$ is an unknown compression of the upper right submatrix of $C_{0;1,1}$, and the blocks that don't concern us are left blank. At this point we do not need another factorization to get $X_2Y_2^H$; instead we can directly derive the compression $X_2Y_2^H$ of $C_{0;1,1}$ from $X_{1,2}$ and $Y_{1,1}$. Clearly we have

$$\begin{aligned} X_2Y_2^H &= \begin{pmatrix} \frac{1}{\omega^2-\omega^{2l+3}} & \cdots & \frac{1}{\omega^2-\omega^{N+1}} \\ \vdots & \cdots & \vdots \\ \frac{1}{\omega^{2k}-\omega^{2l+3}} & \cdots & \frac{1}{\omega^{2k}-\omega^{N+1}} \end{pmatrix}, \\ X_{1,2}Y_{1,1}^H &= \begin{pmatrix} \frac{1}{\omega^{2k+2}-\omega^{N+3}} & \cdots & \frac{1}{\omega^{2k+2}-\omega^{2(N-l)+1}} \\ \vdots & \cdots & \vdots \\ \frac{1}{\omega^{4k}-\omega^{N+3}} & \cdots & \frac{1}{\omega^{N}-\omega^{2(N-l)+1}} \end{pmatrix} = \frac{1}{\omega^{2k}} X_2Y_2^H. \end{aligned}$$

That means we can get $X_2Y_2^H$ by shifting a subblock of $X_1Y_1^H$. A similar situation holds for the splitting of the $(2,2)$ block of $C$ after the first splitting. For the successive compressions in later splittings, a similar shifting can be used. For splitting with general block sizes the shifting is also similar.

The shifting scheme indicates that, in different levels of divide-and-conquer SSS constructions, the compressions of large blocks and small blocks are related. This can be used to further save the compression cost.

## 5. Performance and numerical experiments.

It is well known that the FFT transformation of the Toeplitz matrix $T$ to a Cauchy-like matrix $C$, and the recovery from the solution $\tilde{x}$ of the Cauchy-like system to the solution $x$ of the Toeplitz system, are stable. In addition, the fast divide-and-conquer construction and the SSS system solver in section 3 are both practically stable as we will see in our numerical results. Thus our overall algorithm is stable in practice. Here no extra steps are needed to stabilize the algorithm as required in some other superfast methods [38].

After the precomputations discussed in sections 4.2 and 4.3 are finished, the cost of each SSS construction is only $O(Np^2)$, where $p$ is the maximum of the off-diagonal ranks of the Cauchy matrix $C_0$. The SSS solver also costs $O(Np^2)$. The total cost is thus no more than $O(N \log N) + O(Np^2)$ flops. The total storage requirement is $O(Np)$. For the convenience of coding, we use an $O(N^2p)$ cost precomputation routine as discussed in subsection 4.2.

A preliminary implementation of our superfast solver in Fortran 90 is available at http://www.math.ucla.edu/˜jxia/work/toep. We did experiments on an Itanium 2 1.4 GHz SGI Altix 350 server with a 64-bit Linux operating system and Intel MKL BLAS. Our method (denoted by `NEW`) is compared to Gaussian elimination with partial pivoting (denoted by `GEPP`) (via the LAPACK linear system solver ZGESV [3]). We consider real $N \times N$ Toeplitz matrices $T$ whose entries are random and uniformly distributed in $[0, 1]$. The right-hand sides $b$ are obtained by $b = Tx$, where the exact solution $x$ has random entries uniformly distributed in $[-1, 1]$. Matrices of size $N = 2^k \times 100$, $k = 2, 3, \ldots$, are considered. These matrices are moderately ill-conditioned. For example, for $N = 2^k \times 100$ with $k = 2, \ldots, 7$, the one-norm condition numbers of the matrices increase from the order of about $10^3$ to $10^6$.

For the `NEW` solver two important parameters are involved, the SSS block size $d$ (Figure 2.1) and the tolerance $\tau$ in the compressions of off-diagonal blocks. Here we use the $\tau$-accurate QR factorization with $\tau$ as a relative tolerance (section 2.2).

Figure 5.1 shows the execution time (in seconds) for `GEPP` and our `NEW` solver with different $d$ and $\tau$. For the `NEW` solver only the time for solving the Cauchy-like

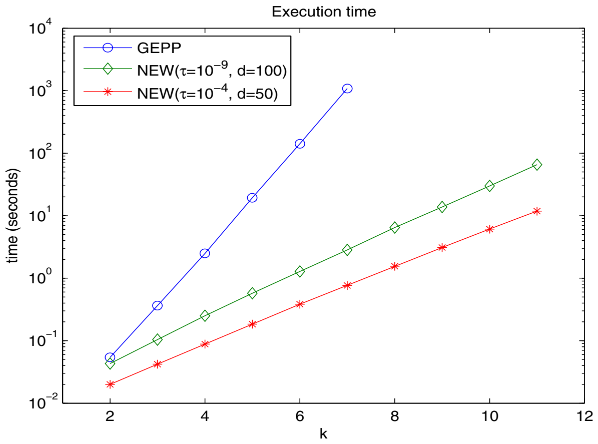


FIG. 5.1. *Computation time (in seconds) versus* $N = 2^k \times 100$*. For the* `NEW` *solver, time for the precomputation is excluded.*

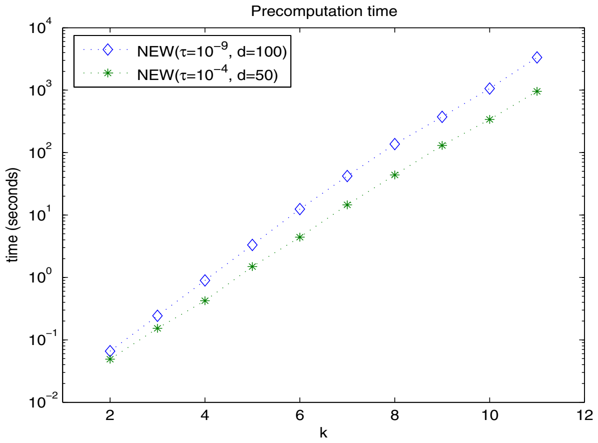


FIG. 5.2. *Time (in seconds) for precomputations in the* `NEW` *solver.*

system (2.3) is reported, as the compressions of off-diagonal blocks can be done in the precomputation, as described in section 4.3, and the SSS construction time is nearly the precomputation time. The precomputation time is shown in Figure 5.2, although the precomputation only needs to be done once for a particular matrix size $N$.

We also consider the *time scaling factor*, that is, the factor by which the time is multiplied when the matrix size doubles; see Figure 5.3. We observe that the time scaling factors for the `NEW` solver are near 2, that is to say the `NEW` solver is close to being a linear time solver.

Figures 5.4 and 5.5 present the errors $\varepsilon_1 = \frac{||\hat{x}-x||_2}{||x||_2}$ and $\varepsilon_2 = \frac{||T\hat{x}-b||_2}{|||T|\,|\hat{x}|+|b|\,||_2}$, respectively, where $\hat{x}$ denotes the numerical solution.

A significant point of the new solver is that we can use a relatively large tolerance $\tau$ in the precomputation and solving, and then use iterative refinement to improve the accuracy. A relatively large $\tau$ leads to relatively small off-diagonal ranks (and thus

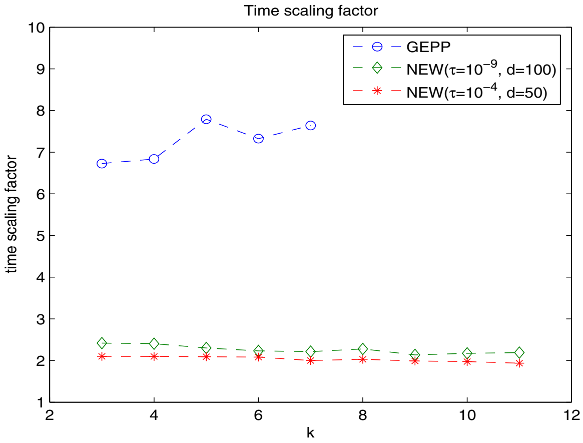


FIG. 5.3. *Time scaling factors.*

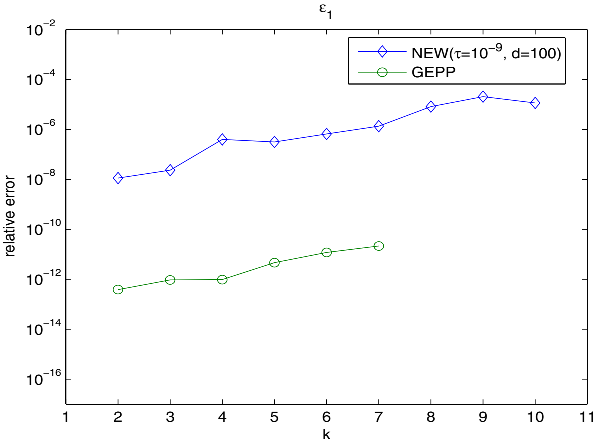


FIG. 5.4. $\varepsilon_1 = \frac{||\hat{x}-x||_2}{||x||_2}$ *versus* $N = 2^k \times 100$.

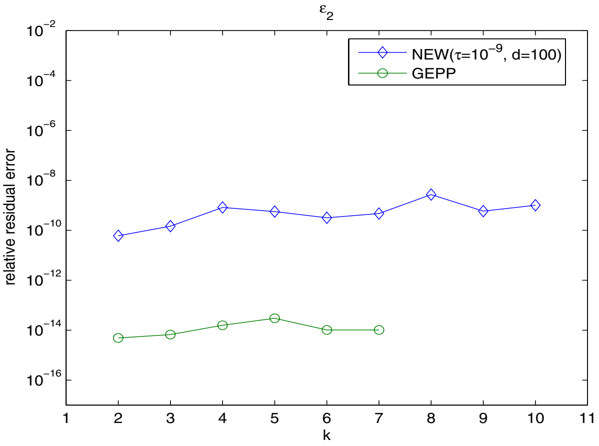


FIG. 5.5. $\varepsilon_2 = \frac{||T\hat{x}-b||_2}{|||T||\hat{x}|+|b|||_2}$ *versus* $N = 2^k \times 100$.

small $p$ in the operation counts above). Iterative refinements are very cheap due to the facts that no precomputation is needed and the solver itself is very fast (Figure 5.1). For $\tau = 10^{-4}$ and $d = 50$ the accuracy results after 2 to 8 steps of iterative refinement are displayed in Figure 5.6. In fact the number of iterative refinement steps required to reach $\varepsilon_2 < 10^{-13}$ is listed in Table 5.1.

Thus it is also clear that our `NEW` solver can also perform well as a preconditioner.

The block size $d$ also affects the performance. Figure 5.7 indicates that for a

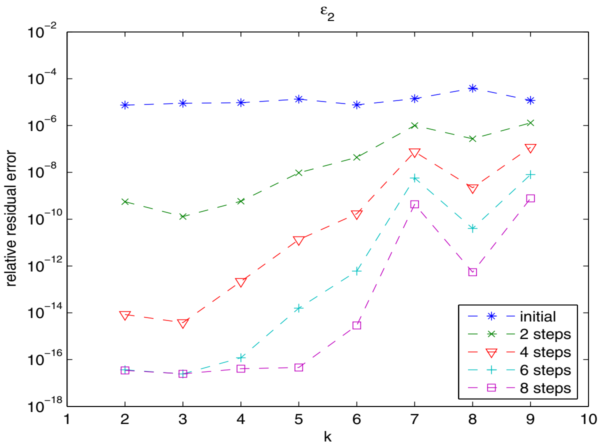


FIG. 5.6. $\varepsilon_2 = \frac{||T\hat{x}-b||_2}{||\,|T|\,|\hat{x}|+|b|\,||_2}$, *initial values (before iterative refinements), and after some steps of iterative refinement.*

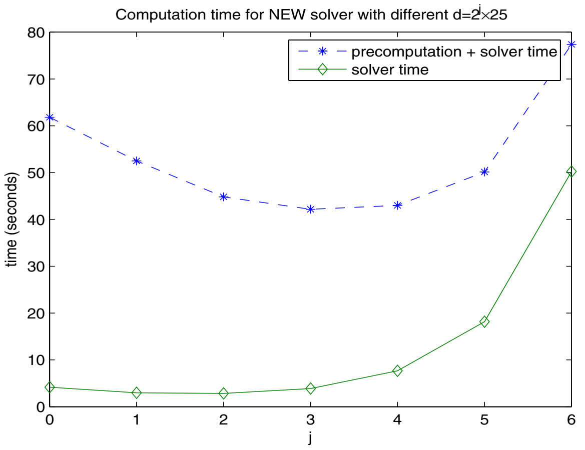


FIG. 5.7. *Computation time of the* `NEW` *solver with* $N = 12800$, $\tau = 10^{-9}$, *and different block size* $d = 2^j \times 25$.

TABLE 5.1
*Number of iterative refinement steps required to reach* $\varepsilon_2 < 10^{-13}$ *for different matrix dimensions* $N = 2^k \times 100$.

| $k$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| Number of steps | 4 | 4 | 5 | 6 | 7 | 15 | 9 | 21 |

given $N$ if $d = 2^j \times 25$ is too large or too small, then both the solver time and the precomputation time can be relatively large. In fact by counting the operations in the algorithm it is possible to determine an optimal $d$; see [12] for the derivation of the optimal choice of $d$ in the SSS solver. We can do similar derivations for both the SSS construction and the SSS solver. We just point out that when $\tau$ gets larger, the off-diagonal ranks decrease and a smaller $d$ should be chosen. In the previous experiments we used two sets of parameters: $(\tau = 10^{-9}, d = 100)$ and $(\tau = 10^{-4}, d = 50)$.

Finally, it turns out that our current preliminary implementation of the solver is slower than the implementation of the algorithm in [38], likely due to our inefficient data structure and nonoptimized codes for SSS matrices. The complicated nature of SSS matrices needs more careful coding and memory management. An improved software implementation is under construction.

## 6. Conclusions and future work.

In this paper we have presented a superfast and practically stable solver for Toeplitz systems of linear equations. A Toeplitz matrix is first transformed into a Cauchy-like matrix, which has a nice low-rank property, and then the Cauchy-like system is solved by a superfast solver. This superfast solver utilizes this low-rank property and makes use of an SSS representation of the Cauchy-like matrix. A fast construction procedure for SSS structures is presented. After a one-time precomputation the solver is very efficient ($O(N \log N) + O(Np^2)$ complexity). Also the algorithm is efficient in that only linear storage is required. In future work we hope to further reduce the precomputation cost and finish a better-designed version of the current Fortran 90 codes in both the computations and the coding.

**Acknowledgments.** Many thanks to the referees for their valuable comments which have greatly improved the development and presentation of this paper. The authors are also grateful to the editors and the editorial assistant for their patience and thoughtful suggestions.

## REFERENCES

[1] G. S. Ammar and W. B. Gragg, *Superfast solution of real positive definite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 61–76.

[2] G. S. Ammar and W. B. Gragg, *Numerical experience with a superfast real Toeplitz solver*, Linear Algebra Appl., 121 (1989), pp. 185–206.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, PA, 1994.

[4] R. R. Bitmead and B. D. O. Anderson, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Linear Algebra Appl., 34 (1980), pp. 103–116.

[5] A. W. Bojanczyk, R. P. Brent, F. R. de Hoog, and D. R. Sweet, *On the stability of the Bareiss and related Toeplitz factorization algorithms*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 40–57.

[6] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximants*, J. Algorithms, 1 (1980), pp. 259–295.

[7] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 349–364.

[8] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 669–686.

[9] T. F. Chan and P. C. Hansen, *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. Signal Process., 40 (1992), pp. 1079–1090.

[10] T. F. Chan and P. C. Hansen, *A look-ahead Levinson algorithm for indefinite Toeplitz systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 490–506.

[11] S. Chandrasekaran and M. Gu, *Fast and stable algorithms for banded plus semiseparable matrices*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 373–384.

[12] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, and D. White, *Fast Stable Solvers for Sequentially Semi-Separable Linear Systems of Equations and Least Squares Problems*, Technical report, University of California, Berkeley, CA, 2003.

[13] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, and D. White, *Some fast algorithms for sequentially semiseparable representations*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 341–364.

[14] S. Chandrasekaran, M. Gu, and W. Lyons, *A Fast and Stable Adaptive Solver for Hierarchically Semi-Separable Representations*, Technical report, UCSB Math 2004-20, University of California, Santa Barbara, CA, 2004.

[15] G. Cybenko, *The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations*, SIAM J. Sci. Statist. Comput., 1 (1980), pp. 303–319.

[16] G. Cybenko, *Error Analysis of Some Signal Processing Algorithms*, Ph.D. thesis, Princeton University, Princeton, NJ, 1978.

[17] F. R. deHoog, *On the solution of Toeplitz systems*, Linear Algebra Appl., 88/89 (1987), pp. 123–138.

[18] P. Dewilde and A. van der Veen, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, MA, 1998.

[19] M. Fiedler, *Hankel and Loewner matrices*, Linear Algebra Appl., 58 (1984), pp. 75–95.

[20] K. A. Gallivan, S. Thirumalai, P. Van Dooren, and V. Vermaut, *High performance algorithms for Toeplitz and block Toeplitz matrices*, Linear Algebra Appl., 241/243 (1996), pp. 343–388.

[21] L. Gemignani, *Schur complements of Bezoutians and the inversion of block Hankel and block Toeplitz matrices*, Linear Algebra Appl., 253 (1997), pp. 39–59.

[22] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1576.

[23] I. Gohberg and V. Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations Operator Theory, 20 (1994), pp. 44–83.

[24] I. Gohberg and V. Olshevsky, *Complexity of multiplication with vectors for structured matrices*, Linear Algebra Appl., 202 (1994), pp. 163–192.

[25] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[26] M. Gu, *Stable and efficient algorithms for structured systems of linear equations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 279–306.

[27] M. Gu and S. C. Eisenstat, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869.

[28] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, in Linear Algebra for Signal Processing, IMA Vol. Math. Appl. 69, Springer, New York, 1995, pp. 63–81.

[29] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Oper. Theory Adv. Appl. 13, Birkhäuser Verlag, Basel, 1984, pp. 109–127.

[30] T. Kailath, *Fredholm resolvents, Wiener-Hopf equations, and Riccati differential equations*, IEEE Trans. Inform. Theory, 15 (1969), pp. 665–672.

[31] T. Kailath, S. Kung, and M. Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407.

[32] T. Kailath and A. H. Sayed, eds., *Fast Reliable Algorithms for Matrices with Structure*, SIAM, Philadelphia, 1999.

[33] P. G. Martinsson, V. Rokhlin, and M. Tygert, *A fast algorithm for the inversion of general Toeplitz matrices*, Comput. Math. Appl., 50 (2005), pp. 741–752.

[34] M. Morf, *Fast Algorithms for Multivariable Systems*, Ph.D. thesis, Department of Electrical Engineering, Stanford University, Stanford, CA, 1974.

[35] B. R. Musicus, *Levinson and Fast Choleski Algorithms for Toeplitz and Almost Toeplitz Matrices*, Technical report, Res. Lab. of Electronics, M.I.T., Cambridge, MA, 1984.

[36] V. Pan, *On computations with dense structured matrices*, Math. Comp., 55 (1990), pp. 179–190.

[37] D. R. Sweet, *The use of pivoting to improve the numerical performance of algorithms for Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 468–493.

[38] M. Van Barel, G. Heinig, and P. Kravanja, *A stabilized superfast solver for nonsymmetric Toeplitz systems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 494–510.

[39] M. Van Barel and P. Kravanja, *A stabilized superfast solver for indefinite Hankel systems*, Linear Algebra Appl., 284 (1998), pp. 335–355.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 4, pp. 1267–1280

# STEEPEST DESCENT AND CONJUGATE GRADIENT METHODS WITH VARIABLE PRECONDITIONING*

ANDREW V. KNYAZEV† AND ILYA LASHUK†

**Abstract.** We analyze the conjugate gradient (CG) method with variable preconditioning for solving a linear system with a real symmetric positive definite (SPD) matrix of coefficients $A$. We assume that the preconditioner is SPD on each step, and that the condition number of the preconditioned system matrix is bounded above by a constant independent of the step number. We show that the CG method with variable preconditioning under this assumption may not give improvement, compared to the steepest descent (SD) method. We describe the basic theory of CG methods with variable preconditioning with the emphasis on "worst case" scenarios, and provide complete proofs of all facts not available in the literature. We give a new elegant geometric proof of the SD convergence rate bound. Our numerical experiments, comparing the preconditioned SD and CG methods, not only support and illustrate our theoretical findings, but also reveal two surprising and potentially practically important effects. First, we analyze variable preconditioning in the form of inner-outer iterations. In previous such tests, the unpreconditioned CG inner iterations are applied to an artificial system with some fixed preconditioner as a matrix of coefficients. We test a different scenario, where the unpreconditioned CG inner iterations solve linear systems with the original system matrix $A$. We demonstrate that the CG-SD inner-outer iterations perform as well as the CG-CG inner-outer iterations in these tests. Second, we compare the CG methods using a two-grid preconditioning with fixed and randomly chosen coarse grids, and observe that the fixed preconditioner method is twice as slow as the method with random preconditioning.

**Key words.** Steepest descent, conjugate gradient, iterative method, inner-outer iterations, variable preconditioning, random preconditioning, preconditioner, condition number, linear systems, circular cone, Householder reflection, convergence rate bound, multigrid

**AMS subject classification.** 65F10

**DOI.** 10.1137/060675290

## 1. Introduction.

Preconditioning, a transformation, usually implicit, of the original linear system aiming at accelerating the convergence of the approximations to the solution, is typically a necessary part of an efficient iterative technique. Modern preconditioning, e.g., based on so-called algebraic multilevel and domain decomposition methods, attempts to become as close to a "black box" ideal of direct solvers as possible. In this attempt, the mathematical structure of the preconditioner, which in the classical case is regarded as some linear transformation, may become very complex, in particular, the linearity can be easily lost, e.g., if the preconditioning itself involves "inner" iterative solvers. The fact that the preconditioner may be nonlinear, or variable, i.e., changing from iteration to iteration, may drastically affect the known theory as well as the practical behavior of preconditioned iterative methods and therefore needs special attention. Our main result is that the conjugate gradient (CG) method with variable preconditioning in certain situations may not give improvement, compared to the steepest descent (SD) method for solving a linear sys-

---

*Received by the editors November 17, 2006; accepted for publication (in revised form) by Q. Ye April 27, 2007; published electronically December 19, 2007. A preliminary version of this paper is available as a technical report [9, v1–v3].

http://www.siam.org/journals/simax/29-4/67529.html

†Department of Mathematical Sciences, University of Colorado at Denver and Health Sciences Center, P.O. Box 173364, Campus Box 170, Denver, CO 80217-3364 (Andrew.Knyazev@cudenver.edu, http://math.cudenver.edu/~aknyazev/, Ilya.Lashuk@cudenver.edu). The first authors' material was based upon work supported by National Science Foundation awards DMS 0208773 and 0612751.

tem with a real symmetric positive definite (SPD) matrix of coefficients. We assume that the preconditioner is SPD on each step, and that the condition number of the preconditioned system matrix is bounded above by a constant.

Let us now introduce the notation, so that we can formulate the main result mathematically. Let $A$ be a real SPD matrix, $(x, y)$ be the standard inner product of real vectors $x$ and $y$, so that $(Ax, y) = (x, Ay)$, and let $\|x\| = \sqrt{(x, x)}$ be the corresponding vector norm. We also use $\|\cdot\|$ to denote the operator norm. The $A$-inner product and the A-norm are denoted by $(x, y)_A = (x, Ay)$ and $\|x\|_A = \sqrt{(x, x)_A}$.

We consider a family of iterative methods to obtain a sequence of approximate solutions $x_k$ of a linear system $Ax = b$ and use the $A$-norm to measure the error $e_k = x - x_k$. The SD and CG methods are well-known iterative procedures that fit into our framework. To accelerate the convergence of the error $e_k$ to zero we introduce preconditioning, i.e., on every iteration $k$ an operator $B_k$, called the preconditioner, possibly different for each iteration $k$, is applied to the residual $r_k = b - Ax_k$. A general algorithm, which includes the preconditioned SD or CG (PSD or PCG, respectively) methods as particular cases, can be presented as follows, e.g., Axelsson [1, p. 540] and Axelsson and Vassilevski [3, Algorithm 5.3]: given $A$, $b$, $\{B_k\}$, $\{m_k\}$, $x_0$, for $k = 0, 1, \ldots$: $r_k = b - Ax_k$, $s_k = B_k^{-1} r_k$, and

$$p_k = s_k - \sum_{l=k-m_k}^{k-1} \frac{(As_k, p_l)}{(Ap_l, p_l)} p_l, \quad x_{k+1} = x_k + \frac{(r_k, p_k)}{(Ap_k, p_k)} p_k, \tag{1.1}$$

where

$$0 \le m_k \le k \text{ and } m_{k+1} \le m_k + 1. \tag{1.2}$$

The latter condition is highlighted in Notay [11, p. 1447, line 1] and ensures that the formula for $p_k$ in (1.1) performs the standard Gram–Schmidt $A$-orthogonalizations to previous search directions, which are already pairwise $A$-orthogonal. The full orthogonalization that performs explicit $A$-orthogonalizations to all previous search directions corresponds to $m_k = k$. Choosing $m_k = \min\{k, 1\}$ gives the PCG method, e.g., described in Golub and Ye [6, IPCG Algorithm]. The connection of this PCG method to the commonly used PCG algorithm is discussed in section 7 following Golub and Ye [6, Remark 2.3]. The shortest recursion $m_k = 0$ leads to the standard PSD method.

It is well known, e.g., D′yakonov [4, p. 34] and Axelsson [1, section 11.1.2, p. 458] that if the preconditioner is SPD and fixed, $B_k = B = B^* > 0$, a preconditioned method, such as (1.1), using the preconditioner $B$ can be viewed as the corresponding unpreconditioned method applied to the preconditioned system $B^{-1}Ax = B^{-1}b$ in the $B$-based inner product $(x, y)_B = (x, By)$. This implies that the theory obtained for unpreconditioned methods remains valid for preconditioned methods, in particular, the $A$-orthogonalization terms with $l < k - 1$ in the sum in (1.1) vanish in exact arithmetic, e.g., Axelsson [1, section 11.2.6, Theorem 11.5]. The situation changes dramatically, however, if different preconditioners $B_k$ are used in the PCG method.

This paper concerns the behavior of method (1.1), where the preconditioner $B_k$ varies from step to step, but remains SPD on each step and the spectral condition number $\kappa\left(B_k^{-1}A\right) = \lambda_{\max}\left(B_k^{-1}A\right)/\lambda_{\min}\left(B_k^{-1}A\right)$ is bounded above by some constant $\kappa_{\max}$ independent of the step number $k$. We note that the matrix $B_k^{-1}A$ is SPD with respect to, e.g., the $B_k$ inner product, so its eigenvalues are real positive. Let us highlight that our assumption $\kappa\left(B_k^{-1}A\right) \le \kappa_{\max}$ can be equivalently written as $\|I - B_k^{-1}A\|_{B_k} \le \gamma$ with $\kappa_{\max} = (1+\gamma)/(1-\gamma)$, assuming without loss of generality that $B_k$ is scaled such that $\lambda_{\max}\left(B_k^{-1}A\right)+\lambda_{\min}\left(B_k^{-1}A\right)=2$. Here, we only deal with methods that are invariant with respect to scaling of $B_k$.

The main result of this paper is that the preconditioned method (1.1) with (1.2) turns into the PSD method with the worst possible convergence rate on every iteration, if the preconditioners $B_k$ satisfying our assumption $\kappa\left(B_k^{-1}A\right)\leq\kappa_{\max}$ are chosen in a special way. We explicitly construct a variable preconditioner that slows down the CG method to the point that the worst linear convergence rate of the SD method is recovered. Thus one can only guarantee that the convergence rate for the method (1.1) with (1.2) is just the same as for the PSD method, $m_k=0$, obtained in Kantorovich [7] and reproduced, e.g., in Kantorovich and Akilov [8, Chapter XV]:

$$\frac{\|e_{k+1}\|_A}{\|e_k\|_A}\leq\frac{\kappa_{\max}-1}{\kappa_{\max}+1}. \tag{1.3}$$

Our proof is geometric and based on the simple fact, proved in section 2, that a nonzero vector multiplied by all SPD matrices with a condition number bounded by a constant generates a pointed circular cone. We apply this fact on every iteration to the current residual vector, which becomes the center of the cone, so all points in the cone correspond to all possible preconditioned residuals. In a somewhat similar way, [6] use the angle between the exact and the perturbed preconditioned residuals. In the CG method context, this cone has a nontrivial intersection with the subspace $A$-orthogonal to all previous search directions. So on each iteration we can choose a preconditioner with the a priori chosen quality, determined by $\kappa_{\max}$, that makes enforcing $A$-orthogonality with respect to all previous search directions useless.

Basic properties of method (1.1), most importantly the local optimality, are derived in section 3. In section 4, we apply our results from section 2 about the cone to obtain a new proof of estimate (1.3). In section 5, we analyze the convergence of the PCG method with variable preconditioning and prove our main result. We assume real arithmetic everywhere in the paper, except for section 6, where we show that our main results also hold for complex Hermitian positive definite matrices. In section 7, we consider two particular PCG algorithms that are often used in practice and describe their behavior with variable preconditioning.

Our numerical experiments in section 8 comparing the preconditioned SD and CG methods support and illustrate our theoretical findings and also reveal some potentially practical important effects. In subsection 8.1, we test the widely used modification of the CG method with a simplified formula for the scalar $\beta_k$ from section 7 and demonstrate that variable preconditioning can make this modification much slower than even the SD method. In subsection 8.2, we analyze inner-outer iterations as variable preconditioning. Finally, in subsection 8.3, we demonstrate that variable preconditioning may surprisingly accelerate the SD and the CG compared to the use of fixed preconditioning in the same methods.

Different aspects of variable preconditioning are considered, e.g., in Axelsson [1] and Axelsson and Vassilevski [2, 3], where rather general nonlinear preconditioning is introduced, and in Golub and Ye [6] and Notay [11], they mainly deal with the case when the preconditioner on each iteration approximates a fixed operator. In Axelsson [1], Axelsson and Vassilevski [2], Golub and Ye [6], and Notay [11], convergence estimates for some iterative methods with variable preconditioning are proved. For recent results and other aspects of variable preconditioning, see Simoncini and Szyld [12, 13, 14] and the references therein. No attempts are apparently made in the literature to obtain a result similar to ours, even though it should appear quite natural and somewhat expected to experts in the area after reading this paper.

## 2. Pointed circular cones represent sets of SPD matrices with varying condition numbers.

For a pair of real nonzero vectors $x$ and $y$ we define the angle between $x$ and $y$ in the usual way as

$$\angle(x,y) = \arccos\left(\frac{(x,y)}{\|x\|\,\|y\|}\right) \in [0,\pi].$$

The following theorem is inspired by Neymeyr [10, Lemma 2.3].

THEOREM 2.1. *The set $\{Cx\}$, where $x$ is a fixed nonzero real vector and $C$ runs through all SPD matrices with condition number $\kappa(C)$ bounded above by some $\kappa_{\max}$, is a pointed circular cone, specifically,*

$$\{Cx : \quad C = C^* > 0,\, \kappa(C) \le \kappa_{\max}\} = \left\{y : \quad \sin\angle\,(x,y) \le \frac{\kappa_{\max}-1}{\kappa_{\max}+1}\right\}.$$

Theorem 2.1 can be proved by constructing our cone as the smallest pointed cone that includes the ball considered in Neymeyr [10, Lemma 2.3]. Preparing for section 6 that deals with the complex case, not covered in Neymeyr [10], we provide a direct proof here based on the following two lemmas. The first lemma is simple and states that the set in question cannot be larger than the cone.

LEMMA 2.2. *Let $x$ be a nonzero real vector and let $C$ be an SPD matrix with spectral condition number $\kappa\,(C)$. Then $\sin\angle\,(x,Cx) \le (\kappa\,(C)-1)/(\kappa\,(C)+1)$.*

*Proof.* Denote $y = Cx$. We have $(x,Cx) = \left(y, C^{-1}y\right) > 0$ since $C$ is SPD, so $y \ne 0$ and $\angle\,(x,y) < \pi/2$. A positive scaling of $C$ and thus of $y$ is obviously irrelevant, so let us choose $y$ to be the orthogonal projection of $x$ onto the one-dimensional subspace spanned by the original $y$. Then from elementary two-dimensional geometry it follows that $\|y-x\| = \|x\|\sin\angle\,(x,y)$. The orthogonal projection of a vector onto a subspace is the best approximation to the vector from the subspace, thus

$$\|x\|\sin\angle\,(x,y) = \|y-x\| \le \|sy-x\| = \|sCx-x\| \le \|sC-I\|\,\|x\|$$

for any scalar $s$, where $I$ is the identity. Taking $s = 2/\left(\lambda_{\max}(C)+\lambda_{\min}(C)\right)$, where $\lambda_{\min}(C)$ and $\lambda_{\max}(C)$ are the minimal and maximal eigenvalues of $C$, respectively, we get $\|sC-I\| = (\kappa\,(C)-1)/(\kappa\,(C)+1)$. ◻

The second lemma implies that every point in the cone can be represented as $Cx$ for some SPD matrix $C$ with $\kappa(C)$ determined by the opening angle of the cone.

LEMMA 2.3. *Let $x$ and $y$ be nonzero real vectors, such that $\angle\,(x,y) \in \left[0,\frac{\pi}{2}\right)$. Then there exists an SPD matrix $C$, such that $Cx = y$ and*

$$\frac{\kappa\,(C)-1}{\kappa\,(C)+1} = \sin\angle\,(x,y)\,.$$

*Proof.* Denote $\alpha = \angle\,(x,y)$. A positive scaling of vector $y$ is irrelevant, so as in the previous proof we choose $y$ to be the orthogonal projection of $x$ onto the one-dimensional subspace spanned by the original $y$, then $\|y-x\| = (\sin\alpha)\,\|x\|$, so the vectors $y-x$ and $(\sin\alpha)x$ are of the same length. This implies that there exists a Householder reflection $H$ such that $H\left((\sin\alpha)\,x\right) = y-x$, cf. Neymeyr [10, Lemma 2.3], so $(I+(\sin\alpha)\,H)\,x = y$. We define $C = I+(\sin\alpha)\,H$ to get $Cx = y$. Any Householder reflection is symmetric and has only two distinct eigenvalues $\pm 1$, so $C$ is also symmetric and has only two distinct positive eigenvalues $1 \pm \sin\alpha$, as $\alpha \in [0,\pi/2)$, and we conclude that $C > 0$ and $\kappa\,(C) = (1+\sin\alpha)/(1-\sin\alpha)$. ◻

**3. Local optimality of the method with variable preconditioning.** Here we discuss some basic properties of method (1.1) with (1.2). We derive a simple, but very useful, error propagation identity in Lemma 3.1. We prove in Lemma 3.2 that the method is well defined and has a certain local $A$-orthogonality property, formulated without a proof in Notay [11, formulas (2.1) and (2.2)] and in the important particular case $m_k = \min\{k, 1\}$ proved in Golub and Ye [6, Lemma 2.1]. Using the local $A$-orthogonality property of Lemma 3.2, we prove the local A-optimality property in Lemma 3.3 by generalizing the result of Golub and Ye [6, Proposition 2.2]. Finally, we derive a trivial Corollary 3.4 from Lemma 3.3, which uses the idea from Golub and Ye [6, p. 1309] of comparison with the PSD method, $m_k = 0$.

The material of this section is inspired by Golub and Ye [6] and may be known to experts in the field, e.g., some even more general facts can be found in Axellson [1, section 12.3.2, Lemma 12.22]. We provide straightforward and complete proofs here suitable for a general audience.

LEMMA 3.1. *Let $A$ and $\{B_k\}$ be SPD matrices. Suppose $p_k$ in method* (1.1) *is well defined and nonzero. Then*

$$e_{k+1} = e_k - \frac{(Ae_k, p_k)}{(Ap_k, p_k)} p_k. \tag{3.1}$$

*Proof.* Recall that $e_k = A^{-1}b - x_k$ and thus $r_k = Ae_k$. Then (3.1) follows immediately from the last formula in (1.1). ☐

LEMMA 3.2. *Let $A$ and $\{B_k\}$ be SPD matrices and $\{m_k\}$ satisfies* (1.2). *Then the error, the preconditioned residual, and the direction vectors generated by method* (1.1) *before the exact solution is obtained are well defined and satisfy*

$$(p_i, p_j)_A = 0,\ k - m_k \le i < j \le k, \tag{3.2}$$

$$(e_{k+1}, s_k)_A = (e_{k+1}, p_i)_A = 0,\ k - m_k \le i \le k. \tag{3.3}$$

*Proof.* We first notice that (3.1) for any $k$ obviously implies

$$(e_{k+1}, p_k)_A = 0. \tag{3.4}$$

For the rest of the proof we use an induction in $k$. Let us take $k = 0$ and suppose $x_0 \neq x$, then $r_0 \neq 0$ and $s_0 \neq 0$ since $B_0$ is SPD. By (1.2), $m_0 = 0$ and thus $p_0 = s_0 \neq 0$, so in the formula for $x_{k+1}$ we do not divide by zero, i.e., $x_{k+1}$ is well defined. There is nothing to prove in (3.2) for $k = 0$ since $m_0 = 0$. Formula (3.4) implies $(e_1, p_0)_A = (e_1, s_0)_A = 0$, i.e., (3.3) holds for $k = 0$. This provides the basis for the induction.

Suppose the statement of the lemma holds for $k - 1$, which is the induction hypothesis, i.e., up to the index $k - 1$ all quantities are well defined and

$$(p_i, p_j)_A = 0,\ k - 1 - m_{k-1} \le i < j \le k - 1, \tag{3.5}$$

$$(e_k, s_{k-1})_A = (e_k, p_i)_A = 0,\ k - 1 - m_{k-1} \le i \le k - 1. \tag{3.6}$$

We now show by contradiction that $x_k \neq x$ implies $p_k \neq 0$. Indeed, if $p_k = 0.2$ then $s_k$ is a linear combination of $p_{k-m_k}, \dots, p_{k-1}$. However, since $m_k \le m_{k-1} + 1$, it follows from (3.6) that

$$(e_k, p_i)_A = 0,\ k - m_k \le i \le k - 1. \tag{3.7}$$

Then we have $(s_k, e_k)_A = 0$. At the same time, since the matrix $B_k^{-1}A$ is $A$-SPD, $s_k = B_k^{-1}Ae_k$ cannot be $A$-orthogonal to $e_k$ unless $s_k = e_k = 0$, i.e., $x_k = x$.

Next, we prove (3.2) by showing that the formula for $p_k$ in (1.1) is a valid step of the Gram–Schmidt orthogonalization process with respect to the $A$-based inner product. If $m_k = 0$, then there is nothing to prove. If $m_k = 1$, then (3.2) gets reduced to $(p_k, p_{k-1})_A = 0$, which follows from the formula for $p_k$ in (1.1). If $m_k \geq 2$, then condition (1.2) implies that vectors $p_{k-m_k}, \ldots, p_{k-1}$ are among the vectors $p_{k-1-m_{k-1}}, \ldots, p_{k-1}$ and therefore are already $A$-orthogonal by the induction assumption (3.5). Then the formula for $p_k$ in (1.1) is indeed a valid step of the Gram–Schmidt orthogonalization process with respect to the $A$-based inner product, so (3.2) holds.

It remains to prove (3.3). We have already established (3.2), and (3.4)–(3.7). Equalities (3.2) and (3.7) imply that $p_k$ and $e_k$ are $A$-orthogonal to $p_{k-m_k}, \ldots, p_{k-1}$. Equality (3.1) implies that $e_{k+1}$ is a linear combination of $e_k$ and $p_k$. Thus, we have $(e_{k+1}, p_i)_A = 0,\ k - m_k \leq i \leq k-1$. Finally, it is enough to notice that $s_k$ is a linear combination of $p_k, p_{k-1}, \ldots, p_{k-m_k}$, so $(e_{k+1}, s_k)_A = 0$. ∎

We now use Lemma 3.2 to prove the local optimality of method (1.1) with (1.2), which generalizes the statement of Golub and Ye [6, Proposition 2.2].

LEMMA 3.3. *Under the assumptions of Lemma* 3.2,

$$\|e_{k+1}\|_A = \min_{p \in span\left\{s_k, p_{k-m_k}, \ldots, p_{k-1}\right\}} \|e_k - p\|_A.$$

*Proof.* We get $e_{k+1} \in e_k + \operatorname{span}\{s_k, p_{k-m_k}, \ldots, p_{k-1}\}$ from the formula for $p_k$ in (1.1) and (3.1). Putting this together with $A$-orthogonality relations (3.3) of the vector $e_{k+1}$ with all vectors that span the subspace finishes the proof. ∎

Two important corollaries follow immediately from Lemma 3.3 by analogy with Golub and Ye [6, Proposition 2.2].

COROLLARY 3.4. *The $A$-norm of the error $\|e_{k+1}\|_A$ in method* (1.1) *with* (1.2) *is bounded above by the $A$-norm of the error of one step of the PSD method, $m_k = 0$, using the same $x_k$ as the initial guess and $B_k$ as the preconditioner, i.e., specifically, $\|e_{k+1}\|_A \leq \min_\alpha \|e_k - \alpha s_k\|_A$.*

COROLLARY 3.5. *Let $m_k > 0$, then the $A$-norm of the error $\|e_{k+1}\|_A$ in method* (1.1) *with* (1.2) *for $k > 0$ satisfies $\|e_{k+1}\|_A \leq \min_{\alpha,\beta} \|e_k - \alpha s_k - \beta(e_k - e_{k-1})\|_A$.*

*Proof.* Under the lemma assumptions, the formula for $p_k$ in (1.1) and (3.1) imply that $e_{k+1} \in e_k + \operatorname{span}\{s_k, p_{k-1}\} = e_k + \operatorname{span}\{s_k, e_k - e_{k-1}\}$, and the $A$-orthogonality relations (3.3) turn into $(e_{k+1}, s_k)_A = 0$ and $(e_{k+1}, p_{k-1})_A = (e_{k+1}, e_k - e_{k-1})_A = 0$, so the vector $e_{k+1}$ is $A$-orthogonal to both vectors that span the subspace. As in the proof of Lemma 3.3, the local $A$-orthogonality implies the local $A$-optimality. ∎

Corollary 3.4 allows us in section 4 to estimate the convergence rate of method (1.1) with (1.2) by comparison with the PSD method, $m_k = 0$,—this idea is borrowed from Golub and Ye [6, p. 1309]. The results of Lemma 3.3 and Corollary 3.5 seem to indicate that an improved convergence rate bound of method (1.1) with (1.2) can be obtained, compared to the PSD method convergence rate bound that follows from Corollary 3.4. Our original intent has been to combine Corollary 3.5 with convergence rate bounds of the heavy ball method, in order to attempt to prove such an improved convergence rate bound. However, our results of section 5 demonstrate that this improvement is impossible under our only assumption $\kappa\left(B_k^{-1}A\right) \leq \kappa_{\max}$, since one can construct such preconditioners $B_k$ which make the minimizing value of $\beta$ in Corollary 3.5 be zero, so Corollary 3.5 gives no improvement compared to Corollary 3.4.

## 4. Convergence rate bounds for variable preconditioning.

The classical Kantorovich and Akilov [8, Chapter XV] convergence rate bound (1.3) for the PSD method is "local" in the sense that it relates the $A$-norm of the error on two subsequent iterations and does not depend on previous iterations. Thus, it remains valid when the preconditioner $B_k$ changes from iteration to iteration, while the condition number $\kappa\left(B_k^{-1}A\right)$ is bounded above by some constant $\kappa_{\max}$ independent of $k$. The goal of this section is to give an apparently new simple proof of the estimate (1.3) for the PSD method, based on our cone Theorem 2.1, and to extend this statement to cover the general method (1.1) with (1.2), using Corollary 3.4.

We denote the angle between two real nonzero vectors with respect to the $A$-based inner product by

$$\angle_A(x,y) = \arccos\left(\frac{(x,y)_A}{\|x\|_A\,\|y\|_A}\right) \in [0,\pi]$$

and express the error reduction ratio for the PSD method in terms of the angle with respect to the $A$-based inner product.

Lemma 4.1. *On every step of the PSD algorithm,* (1.1) *with* $m_k = 0$*, the error reduction factor takes the form* $\|e_{k+1}\|_A/\|e_k\|_A = \sin(\angle_A(e_k, B_k^{-1}Ae_k))$.

*Proof.* By (3.3), we have $(e_{k+1}, p_k)_A = 0$. Now, for $m_k = 0$, in addition, $p_k = s_k$, so $0 = (e_{k+1}, p_k)_A = (e_{k+1}, s_k)_A = (e_{k+1}, x_{k+1} - x_k)_A$, i.e., the triangle with vertices $x$, $x_k$, $x_{k+1}$ is right-angled in the $A$-inner product, where the hypotenuse is $e_k = x - x_k$. Therefore, $\|e_{k+1}\|_A / \|e_k\|_A = \sin(\angle_A(e_k, x_{k+1} - x_k)) = \sin(\angle_A(e_k, s_k))$, where $s_k = B_k^{-1}(b - Ax_k) = B_k^{-1}Ae_k$ by (1.1). ☐

Let us highlight that Lemma 4.1 provides an exact expression for the error reduction factor, not just a bound—we need this in the proof of Theorem 5.1 in the next section. Combining the results of Lemmas 2.2 and 4.1 together immediately leads to (1.3) for the PSD method, where $m_k = 0$. Finally, taking into account Corollary 3.4, by analogy with the arguments of Golub and Ye [6, p. 1309] and decrypting a hidden statement in Golub and Ye [6, Lemma 3.5], we get the following theorem.

Theorem 4.2. *Convergence rate bound* (1.3) *holds for method* (1.1) *with* (1.2).

## 5. The convergence rate bound is sharp.

Here we formulate and prove the main result of the paper that one can only guarantee the convergence rate described by (1.3) for method (1.1) with (1.2) with variable preconditioning if one only assumes $\kappa\left(B_k^{-1}A\right) \le \kappa_{\max}$. Let us remind the reader that (1.3) also describes the convergence rate for the PSD method, (1.1) with $m_k = 0$. We now show that adding more vectors to the PSD iterative recurrence results in no improvement in convergence, if a specially constructed set of variable preconditioners is used.

Theorem 5.1. *Let an SPD matrix $A$, vectors $b$ and $x_0$, and $\kappa_{\max} > 1$ be given. Assuming that the matrix size is larger than the number of iterations, one can choose a sequence of SPD preconditioners $B_k$, satisfying $\kappa(B_k^{-1}A) \le \kappa_{\max}$, such that method* (1.1) *with* (1.2) *turns into the PSD method,* (1.1) *with* $m_k = 0$*, and on every iteration*

$$\frac{\|e_{k+1}\|_A}{\|e_k\|_A} = \frac{\kappa_{\max} - 1}{\kappa_{\max} + 1}. \tag{5.1}$$

*Proof.* We construct the sequence $B_k$ by induction. First, we choose any vector $q_0$, such that $\sin\angle_A(q_0, e_0) = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$. According to Lemma 2.3 applied in the $A$-inner product, there exists an $A$-SPD matrix $C_0$ with condition number $\kappa(C_0) = \kappa_{\max}$, such that $C_0 e_0 = q_0$. We define the SPD $B_0 = AC_0^{-1}$, then $\kappa(B_0^{-1}A) =$

$\kappa(C_0) = \kappa_{\max}$. We have $s_k = B_k^{-1}Ae_k$, so such a choice of $B_0$ implies $s_0 = q_0$. Also, we have $p_0 = s_0$, i.e., the first step is always a PSD step; thus, by Lemma 4.1 we have proved (5.1) for $k = 0$. Note that $(e_1, p_0)_A = 0$ by (3.3).

Second, we make the induction assumption: let preconditioners $B_l$ for $l \leq k-1$ be constructed, such that $\|e_{l+1}\|_A/\|e_l\|_A = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$ and $(e_k, p_l)_A = 0$ hold for all $l \leq k-1$. The dimension of the space is greater than the total number of iterations by our assumption, so there exists a vector $u_k$, such that $(u_k, p_l)_A = 0$ for $l \leq k-1$ and $u_k$ and $e_k$ are linearly independent. Then the two-dimensional subspace spanned by $u_k$ and $e_k$ is $A$-orthogonal to $p_l$ for $l \leq k-1$.

Let us consider the boundary of the pointed circular cone made of vectors $q_k$ satisfying the condition $\sin\angle_A(q_k, e_k) = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$. This conical surface has a nontrivial intersection with the 2D subspace spanned by $u_k$ and $e_k$, since $e_k$ is the cone axis. Let us choose vector $q_k$ in the intersection, This vector will be obviously $A$-orthogonal to $p_l$, $l \leq k-1$.

Applying the same reasoning as for constructing $B_0$, we deduce that there exists an SPD $B_k$ such that $\kappa(B_k^{-1}A) \leq \kappa_{\max}$ and $B_k^{-1}Ae_k = q_k$. With such a choice of $B_k$ we have $s_k = q_k$. Since $q_k = s_k$ is $A$-orthogonal to $p_l$ for all $l \leq k-1$, it turns out that $p_k = s_k$, no matter how $\{m_k\}$ are chosen. This means that $x_{k+1}$ is obtained from $x_k$ by a steepest descent step. Then we apply Lemma 4.1 and conclude that (5.1) holds. We note, that $(e_{k+1}, p_l)_A = 0$ for all $l \leq k$. Indeed, $(e_{k+1}, p_l)_A = 0$ for all $l \leq k-1$ since $e_{k+1}$ is a linear combination of $e_k$ and $p_k = s_k = q_k$, both $A$-orthogonal to $p_l$ for $l \leq k-1$. Finally, $(e_{k+1}, p_k)_A = 0$ by (3.3). This completes the construction of $\{B_k\}$ by induction and thus the proof. □

Let us highlight that the statement of Theorem 5.1 consists of two parts: first, it is possible to have the PCG method with variable preconditioning that converges no faster than the PSD method with the same preconditioning; and second, moreover, it is possible that the PCG method with variable preconditioning converges no faster than the worst possible theoretical convergence rate for the PSD method described by (1.3). Numerical tests in section 8 show that the former possibility is more likely than the latter. Specifically, we demonstrate numerically in subsection 8.3 that the PCG and PSD methods with random preconditioning converge with the same speed, but both are much faster than what bound (1.3) predicts.

## 6. Complex Hermitian case.

In all other sections of this paper we assume for simplicity that matrices and vectors are real. However, our main results also hold when matrices $A$ and $\{B_k\}$ are complex Hermitian positive definite. Here we discuss modifications to statements and proofs in sections 2, 4, and 5 in order to cover the complex Hermitian case, assuming the scalar product be linear in its first argument and conjugate-linear in its second argument, as usual.

In section 2, the first thing to be changed is the definition of the angle between two nonzero vectors $x, y \in \mathbb{C}^n$, where an absolute value is now taken,

$$\angle(x, y) = \arccos\left|\frac{(x, y)}{\|x\|\,\|y\|}\right| \in \left[0, \frac{\pi}{2}\right],$$

that makes the angle acute and invariant with respect to complex nonzero scaling of the vectors. Lemma 2.2 remains valid in the complex case.

Lemma 6.1. *Let $x$ be a nonzero complex vector, and $C$ be a complex Hermitian positive definite matrix with the spectral condition number $\kappa(C)$, then* $\sin\angle(x, Cx) \leq (\kappa(C) - 1)/(\kappa(C) + 1)$.

*Proof.* Denote $y = Cx$ and let $\gamma = (x, y) / \|y\|^2$, then $\gamma y$ is the projection of $x$ onto span$\{y\}$, and $\angle(x, y) = \angle(x, \gamma y)$. Moreover, $(x, \gamma y) = (x, y)\bar{\gamma} = (x, y)(y, x)/\|y\|^2$ is real—we need this fact later in the proof of Lemma 6.2. We redefine $y$ to $\gamma y$. The rest of the proof is exactly the same as that of Lemma 2.2, since the identity $\|y - x\| = \|x\| \sin \angle(x, y)$, where $y$ is scaled by a complex scalar to be the orthogonal projection of $x$ onto span$\{y\}$, still holds in the complex case with the new definition of the angle. ☐

Lemma 2.3 and, thus, Theorem 2.1 do not hold in the complex case after the straightforward reformulation. A trivial counterexample is a pair of vectors $x \neq 0$ and $y = ix$—the angle between $x$ and $y$ is obviously zero, yet it is impossible that $y = Cx$ for any complex Hermitian matrix $C$, since the inner product $(x, y) = -i\|x\|^2$ is not a real number. This counterexample also gives an idea for a simple fix.

LEMMA 6.2. *Let $x$ and $y$ be nonzero complex vectors, such that $\angle(x, y) \neq \pi/2$. Then there exists a complex Hermitian positive definite matrix $C$ and a complex scalar $\gamma$, such that $Cx = \gamma y$ and $(\kappa(C) - 1)/(\kappa(C) + 1) = \sin \angle(x, y)$.*

*Proof.* We first scale the complex vector $y$ as in the proof of Lemma 6.1 to make $y$ be the projection of $x$ onto span$\{y\}$. The rest of the proof is similar to that of Lemma 2.3, but we have to be careful working with the Householder reflection in the complex case, so we provide the complete proof.

The redefined $y$ is the projection of $x$ onto span$\{y\}$, thus, $\|y - x\| = (\sin \alpha)\|x\|$, so the vectors $u = y - x$ and $v = (\sin \alpha)x$ are of the same length. Moreover, their inner product $(u, v)$ is real, since $(x, y)$ is real; see the proof of Lemma 6.1. This implies that the Householder reflection $Hz = z - 2(w, z)w$, where $w = (u - v)/\|u - v\|$, acts on $z = u$ such that $Hu = v$, i.e., $H((\sin \alpha)x) = y - x$, so $(I + (\sin \alpha)H)x = y$. We define $C = I + (\sin \alpha)H$ to get $Cx = y$.

The Householder reflection $H$ is Hermitian and has only two distinct eigenvalues $\pm 1$, so $C$ is also Hermitian and has only two distinct positive eigenvalues $1 \pm \sin \alpha$, as $\alpha \in [0, \pi/2)$, and we conclude that $C > 0$ and $\kappa(C) = (1 + \sin \alpha)/(1 - \sin \alpha)$. ☐

The same change then makes Theorem 2.1 work in the complex case.

THEOREM 6.3. *The set $\{\gamma Cx\}$, where $x$ is a fixed nonzero complex vector, $\gamma$ runs through all nonzero complex scalars, and $C$ runs through all complex Hermitian positive definite matrices with condition number $\kappa(C)$ bounded above by some $\kappa_{\max}$, is a pointed circular cone, specifically,*

$$\{\gamma\, Cx : \gamma \neq 0, C = C^* > 0,\ \kappa(C) \leq \kappa_{\max}\} = \left\{ y : \sin \angle(x, y) \leq \frac{\kappa_{\max} - 1}{\kappa_{\max} + 1} \right\}.$$

Section 3 requires no changes other then replacing "SPD" with "Hermitian positive definite." In section 4, we just change the definition of the $A$-angle to

$$\angle_A(x, y) = \arccos \left| \frac{(x, y)_A}{\|x\|_A \|y\|_A} \right| \in \left[0, \frac{\pi}{2}\right],$$

and then Lemma 4.1 holds without any further changes.

Finally, the statement of Theorem 5.1 from section 5 allows for a straightforward generalization.

THEOREM 6.4. *Let a Hermitian positive definite matrix $A$, complex vectors $b$ and $x_0$, and $\kappa_{\max} > 1$ be given. Assuming that the matrix size is larger than the number of iterations, one can choose a sequence of Hermitian positive definite preconditioners $B_k$, satisfying $\kappa(B_k^{-1}A) \leq \kappa_{\max}$, such that method* (1.1) *with* (1.2) *turns into the PSD* *method,* (1.1) *with* $m_k = 0$*, and on every iteration*

$$\frac{\|e_{k+1}\|_A}{\|e_k\|_A} = \frac{\kappa_{\max} - 1}{\kappa_{\max} + 1}. \tag{6.1}$$

*Proof.* Only a small change in the proof of Theorem 5.1 is needed. We first choose any vector $q_0'$, satisfying $\sin \angle_A(q_0, e_0) = (\kappa_{\max} - 1)/(\kappa_{\max} + 1)$. Then by Lemma 6.2 we obtain the complex Hermitian positive definite matrix $C_0$ and the complex scalar $\gamma$ such that $C_0 e_0 = \gamma q_0'$. Finally, we choose $q_0$ to be $\gamma q_0'$ and continue as in the proof of Theorem 5.1. The same modification is made in the choice of the vectors $q_k$ for $k \geq 1$ later in the proof. ☐

**7. Practical PCG algorithms.** In this section we briefly discuss two particular well-known PCG algorithms that are often used in practice. Our discussion here is motivated by and follows Golub and Ye [6, Remark 2.3]. Suppose $A$, $b$, $x_0$, $r_0 = b - Ax_0$, $\{B_k\}$ for $k = 0, 1, \ldots$ are given, and consider Algorithm 7.1 where $\beta_k$ on line 7.1 is defined either by expression

$$\beta_k = \frac{(s_k, r_k)}{(s_{k-1}, r_{k-1})}, \tag{7.1}$$

or by expression

$$\beta_k = \frac{(s_k, r_k - r_{k-1})}{(s_{k-1}, r_{k-1})}. \tag{7.2}$$

Formula (7.1) is more often used in practice compared to (7.2), since it can be implemented in such a way that does not require storing the extra vector $r_{k-1}$.

**Algorithm 7.1**

```
1: for $k = 0, 1, \ldots$ do
2:    $s_k = B_k^{-1} r_k$
3:    if $k = 0$ then
4:       $p_0 = s_0$
5:    else
6:       $p_k = s_k + \beta_k p_{k-1}$ (where $\beta_k$ is defined by either (7.2) or (7.1) for all iterations)
7:    end if
8:    $\alpha_k = \frac{(s_k, r_k)}{(p_k, Ap_k)}$
9:    $x_{k+1} = x_k + \alpha_k p_k$
10:   $r_{k+1} = r_k - \alpha_k A p_k$
11: end for
```

If the preconditioner is SPD and fixed, it is well known, e.g., Golub and Ye [6, Remark 2.3], that $(s_k, r_{k-1}) = 0$, so formula (7.2) coincides with (7.1) and Algorithm 7.1 is described by (1.1) with $m_k = \min(k, 1)$. Of course, in this case the choice $m_k = \min(k, 1)$ is enough to keep *all* search directions $A$-orthogonal in exact arithmetic.

Things become different when variable preconditioning is used. It is well known, e.g., Golub and Ye [6, Remark 2.3] and Notay [11, Table 2], that using formula (7.1) for $\beta_k$ can significantly slow down the convergence, and we provide our own numerical evidence of that in section 8. At the same time, comparing Lemma 3.2

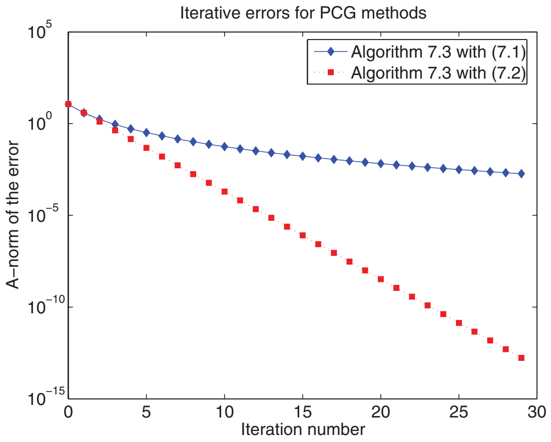


FIG. 8.1. *Algorithm* 7.1 *with* (7.1) *fails to provide the PSD convergence rate.*

with Lemma 2.1 from Golub and Ye [6], we can show, see Knyazev and Lashuk [9, v1], that Algorithm 7.1 with $\beta_k$ defined by (7.2), which is exactly as stated by Golub and Ye [6, IPCG Algorithm], is equivalent to the particular case of (1.1), namely with $m_k = \min(k, 1)$, and therefore is guaranteed by Theorem 4.2 to converge with at least the same speed as the PSD method.

## 8. Numerical experiments.

We first illustrate the main theoretical results of the paper numerically for a model problem. We numerically investigate the influence of the choice for $\beta_k$ between formulas (7.1) and (7.2) in Algorithm 7.1 and observe that (7.2) leads to the theoretically predicted convergence rate, while (7.1) may significantly slow down the convergence. Second, we test the convergence of inner-outer iteration schemes, where the inner iterations play the role of the variable preconditioning in the outer PCG iteration, and we illustrate our main conclusion that variable preconditioning may effectively reduce the convergence speed of the PCG method to the speed of the PSD method. Third, and last, we test the PSD and PCG methods with preconditioners of the same quality chosen randomly. We observe a surprising acceleration of the PCG method compared to the use of only one fixed preconditioner; at the same time, we show that the PSD method with random preconditioners works as well as the PCG method, which explains the PCG acceleration and again supports our main conclusion.

### 8.1. Numerical illustration of the main results.

Here, we use the standard 3-point approximation of the one-dimensional Laplacian of the size 200 as the matrix $A$ of the system. To simulate the application of the variable preconditioner, we essentially repeat the steps described in the proof of Theorem 5.1, i.e., we fix the condition number $\kappa\left(B_k^{-1}A\right) = 2$ and on each iteration we generate a pseudorandom vector $s_k$, which is $A$-orthogonal to previous search directions and such that the $A$-angle between $s_k$ and $e_k$ satisfies $\sin\left(\angle_A\left(s_k, e_k\right)\right) = (\kappa - 1)/(\kappa + 1)$.

We summarize the numerical results of this subsection on Figure 8.1, where the horizontal axis represents the number of iterations and the vertical axis represents the $A$-norm of the error. The iteration count actually starts from 1, so the $A$-norm of the error on the 0th iteration $\|e_0\|_A$ is just the $A$-norm of the initial error. The straight dotted (red in the electronic version) line marked with squares on Figure 8.1

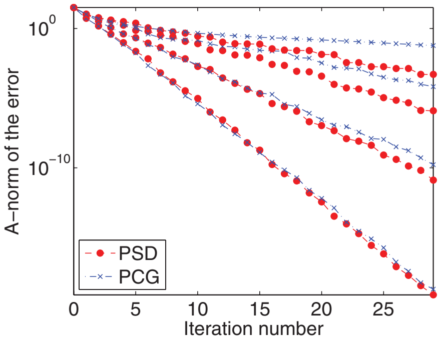


FIG. 8.2. *The PSD and PCG methods with preconditioning by inner CG with different stopping criteria* $\eta = 0.2, 0.4, 0.6,$ *and* $0.8$ *(from the bottom to the top).*

represents the PSD theoretical bound (1.3) and at the same time it perfectly coincides, which illustrates the statements of Theorem 5.1, with the change of the $A$-norm of the error in the case where the complete $A$-orthogonalization is performed, i.e., $m_k = k$ in method (1.1), as well as in the case where Algorithm 7.1 with $\beta_k$ defined by (7.2) is used. The curved solid (blue) line marked with diamonds represents the convergence of Algorithm 7.1 with $\beta_k$ defined by (7.1), which visibly performs much worse in this test compared to Algorithm 7.1 with (7.2). Notay's paper [11, section 5.2] contains analogous results comparing the change in the convergence rate using formulas (7.1) and (7.2), but it misses a comparison with the PSD method. To check our results of section 6, we repeat the tests in the complex arithmetic. The figure generated is similar to Figure 8.1, so we do not reproduce it here.

**8.2. Inner-outer iterations as variable preconditioning.** Inner-outer iterative schemes, where the inner iterations play the role of the variable preconditioner in the outer PCG iteration is a traditional example of variable preconditioning; see, e.g., Golub and Ye [6] and Notay [11]. Previously published tests analyze an approximation of some fixed preconditioner, $B_k \approx B$, different from $A$, by inner iterations, typically using the PCG method. The quality of the approximation is determined by the stopping criteria of the inner PCG method. A typical conclusion is that the performance of the outer PCG method improves and starts behaving like the PCG method with the fixed preconditioner $B$ when $B_k$ approximates $B$ more accurately by performing more inner iterations.

The idea of our tests in this subsection is different: we approximate $B_k \approx B = A$. The specific setup is the following. We take a diagonal matrix $A$ with all integer entries from 1 to 2000, with the right-hand side zero and a random normally distributed zero mean initial guess; we do the same for the PSD and PCG methods. For preconditioning on the $k$th step, applied to the residual $r_k$, we run the standard CG method without preconditioning as inner iterations, using the zero initial approximation, and for the stopping criteria we compute the norm of the true residual at every inner iteration and iterate until it gets smaller than $\eta\|r_k\|$ for a given constant $\eta$. On Figure 8.2, we demonstrate the performance of the PSD and PCG methods for four values of $\eta = 0.2, 0.4, 0.6,$ and $0.8$ (from the bottom to the top). We observe that the PSD, displayed using dashed (red in the electronic version) lines marked with circles and PCG shown as dash-dot (blue) lines with x-marks methods both converge with a similar rate, for each tested value of $\eta$. We notice here that the PSD method is even

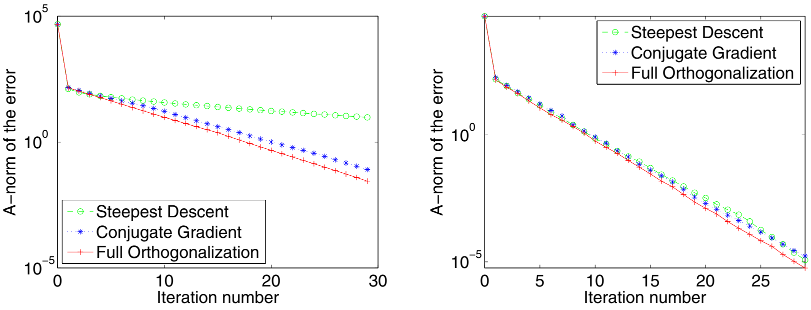


FIG. 8.3. *Two-grid preconditioning with fixed (left) and random (left) coarse grids.*

a bit faster than the PCG method. This does not contradict our Corollary 3.4, since the preconditioners $B_k$ here are evidently different in the PSD and PCG methods even though they are constructed using the same principle.

**8.3. Random vs. fixed preconditioning.** In this subsection, we numerically investigate a situation where random preconditioners of a similar quality are used in the course of iterations. The system matrix is the standard 3-point finite-difference approximation of the one-dimensional Laplacian using 3000 uniform mesh points and the Dirichlet boundary conditions. We test the simplest multigrid preconditioning using two grids, where the number of coarse grid points is 600. We set the interpolation to be linear, the restriction to be the transpose of the interpolation, the coarse-grid operator to be defined by the Galerkin condition, and the smoother to be the Richardson iteration. On Figure 8.3 (left), we once choose (pseudo-)randomly 600 coarse mesh points and build the fixed two-grid preconditioner, based on this choice. On Figure 8.3 (right), we choose 600 new random coarse mesh points and rebuild the two-grid preconditioner on each iteration. We note that in the algebraic multigrid the geometric information about the actual position of the coarse grid points is not available, so the random choice of the coarse grids may be an interesting alternative to traditional approaches.

Figure 8.3 displays the convergence history for the PSD (top), PCG (middle), and PCG with the full orthogonalization (bottom) with the same random initial guess using the fixed (left) and variable (right) two-grid preconditioners. On Figure 8.3 (left), for a fixed preconditioner, we observe the expected convergence behavior, with the PSD being noticeably the slowest and the PCG with the full orthogonalization being slightly faster than the standard PCG. Figure 8.3 (right) demonstrates that all three methods with the variable random preconditioner converge with essentially the same rate, which again illustrates the main result of the paper that the PCG method with variable preconditioning may just converge with the same speed as the PSD method.

Figure 8.3 reveals a surprising fact that the methods with random preconditioning converge twice as fast as the methods with fixed preconditioning! We highlight that Figure 8.3 shows a typical case, not a random outlier, as we confirm by repeating the fixed preconditioner test in the left panel for *every* random preconditioner used in the right panel of Figure 8.3 and by running the tests multiple times with different seeds. Our informal explanation for the fast convergence of the PSD method with random preconditioning is based on Lemma 4.1 that provides the exact expression for the error reduction factor as $\sin(\angle_A(e_k, B_k^{-1}Ae_k))$. It takes its largest value only if $e_k$ is one of specific linear combination of the eigenvectors of $B_k^{-1}Ae$ corresponding to the two extreme eigenvalues. If $B_k$ is fixed, the error $e_k$ in the PSD method after several first iterations approaches these magic linear combinations, e.g., Forsythe [5], and the convergence rate reaches its upper bound. If $B_k$ changes randomly, as in our test, the average "effective" angle is smaller, i.e., the convergence is faster.

**Conclusions.** We use geometric arguments to investigate the behavior of the PCG methods with variable preconditioning under a rather weak assumption that the quality of the preconditioner is fixed. Our main result is negative in its nature: We show that under this assumption the PCG method with variable preconditioning may converge as slow as the PSD method, moreover, as the PSD method with the slowest rate guaranteed by the classical convergence rate bound. In particular, which gives the negative answer, under our assumption, to the question asked in Golub and Ye [6, section 6, Conclusion] whether better bounds for the steepest descent reduction factor may exist for Algorithm 7.1 with (7.2).

Stronger assumptions on variable preconditioning, e.g., such as made in Golub and Ye [6] and Notay [11] that the variable preconditioners are all small perturbations of some fixed preconditioner, are necessary in order to hope to prove a convergence rate bound of the PCG method with variable preconditioning resembling the standard convergence rate bound of the PCG method with fixed preconditioning. Such stronger assumptions hold in many presently known real life applications of the PCG methods with variable preconditioning, but often require extra computational work, e.g., more inner iterations in the inner-outer iterative methods.

## REFERENCES

[1] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, Cambridge, UK, 1994.

[2] O. Axelsson and P. S. Vassilevski, *A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 625–644.

[3] O. Axelsson and P. S. Vassilevski, *Variable-step multilevel preconditioning methods.* I. *Self-adjoint and positive definite elliptic problems*, Numer. Linear Algebra Appl., 1 (1994), pp. 75–101.

[4] E. G. D′yakonov, *Optimization in Solving Elliptic Problems*, CRC Press, Boca Raton, FL, 1996.

[5] G. E. Forsythe, *On the asymptotic directions of the s-dimensional optimum gradient method*, Numer. Math., 11 (1968), pp. 57–76.

[6] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput., 21 (1999/00), pp. 1305–1320.

[7] L. V. Kantorovič, *On the method of steepest descent*, Doklady Akad. Nauk SSSR (N.S.), 56 (1947), pp. 233–236.

[8] L. V. Kantorovich and G. P. Akilov, *Functional Analysis in Normed Spaces*, Pergamon Press, New York, 1964.

[9] A. V. Knyazev and I. Lashuk, *Steepest Descent and Conjugate Gradient Methods with Variable Preconditioning*, Electronic. math.NA/0605767, arXiv.org, available online at http://arxiv.org/abs/math/0605767, 2006–2007.

[10] K. Neymeyr, *A geometric theory for preconditioned inverse iteration.* I. *Extrema of the Rayleigh quotient*, Linear Algebra Appl., 322 (2001), pp. 61–85.

[11] Y. Notay, *Flexible conjugate gradients*, SIAM J. Sci. Comput., 22 (2000), pp. 1444–1460.

[12] V. Simoncini and D. B. Szyld, *Flexible inner-outer Krylov subspace methods*, SIAM J. Numer. Anal., 40 (2002), pp. 2219–2239.

[13] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.

[14] V. Simoncini and D. B. Szyld, *On the occurrence of superlinear convergence of exact and inexact Krylov subspace methods*, SIAM Rev., 47 (2005), pp. 247–272.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1281–1296

# PAGERANK COMPUTATION, WITH SPECIAL ATTENTION TO DANGLING NODES*

ILSE C. F. IPSEN† AND TERESA M. SELEE†

**Abstract.** We present a simple algorithm for computing the PageRank (stationary distribution) of the stochastic Google matrix $G$. The algorithm lumps all dangling nodes into a single node. We express lumping as a similarity transformation of $G$ and show that the PageRank of the nondangling nodes can be computed separately from that of the dangling nodes. The algorithm applies the power method only to the smaller lumped matrix, but the convergence rate is the same as that of the power method applied to the full matrix $G$. The efficiency of the algorithm increases as the number of dangling nodes increases. We also extend the expression for PageRank and the algorithm to more general Google matrices that have several different dangling node vectors, when it is required to distinguish among different classes of dangling nodes. We also analyze the effect of the dangling node vector on the PageRank and show that the PageRank of the dangling nodes depends strongly on that of the nondangling nodes but not vice versa. Last we present a Jordan decomposition of the Google matrix for the (theoretical) extreme case when all Web pages are dangling nodes.

**Key words.** stochastic matrix, stationary distribution, lumping, rank-one matrix, power method, Jordan decomposition, similarity transformation, Google

**AMS subject classifications.** 65F10, 65F50, 65C40, 15A06, 15A18, 15A21, 15A51, 68P20

**DOI.** 10.1137/060664331

## 1. Introduction.

The order in which the search engine Google displays the Web pages is determined, to a large extent, by the *PageRank* vector [7, 33]. The PageRank vector contains, for every Web page, a ranking that reflects the importance of the Web page. Mathematically, the PageRank vector $\pi$ is the stationary distribution of the so-called *Google matrix*, a sparse stochastic matrix whose dimension exceeds 11.5 billion [16]. The Google matrix $G$ is a convex combination of two stochastic matrices

$$G = \alpha S + (1-\alpha)E, \qquad 0 \leq \alpha < 1,$$

where the matrix $S$ represents the link structure of the Web, and the primary purpose of the rank-one matrix $E$ is to force uniqueness for $\pi$. In particular, element $(i, j)$ of $S$ is nonzero if Web page $i$ contains a link pointing to Web page $j$.

However, not all Web pages contain links to other pages. Image files or pdf files, and uncrawled or protected pages have no links to other pages. These pages are called *dangling nodes*, and their number may exceed the number of nondangling pages [11, section 2]. The rows in the matrix $S$ corresponding to dangling nodes would be zero if left untreated. Several ideas have been proposed to deal with the zero rows and force $S$ to be stochastic [11]. The most popular approach adds artificial links to the dangling nodes, by replacing zero rows in the matrix with the same vector, $w$, so that the matrix $S$ is stochastic.

It is natural as well as efficient to exclude the dangling nodes with their artificial links from the PageRank computation. This can be done, for instance, by

---

[*] Received by the editors July 5, 2006; accepted for publication (in revised form) by D. Boley May 7, 2007; published electronically December 19, 2007.
http://www.siam.org/journals/simax/29-4/66433.html

[†] Department of Mathematics, North Carolina State University, P.O. Box 8205, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, tmselee@ncsu.edu, http://www4.ncsu.edu/~ipsen, http://www4.ncsu.edu/~tmselee).

"lumping" all the dangling nodes into a single node [32]. In section 3, we provide a rigorous justification for lumping the dangling nodes in the Google matrix $G$, by expressing lumping as a similarity transformation of $G$ (Theorem 3.1). We show that the PageRank of the nondangling nodes can be computed separately from that of the dangling nodes (Theorem 3.2), and we present an efficient algorithm for computing PageRank by applying the power method only to the much smaller, lumped matrix (section 3.3). Because the dangling nodes are excluded from most of the computations, the operation count depends, to a large extent, on only the number of nondangling nodes, as opposed to the total number of Web pages. The algorithm has the same convergence rate as the power method applied to $G$, but is much faster because it operates on a much smaller matrix. The efficiency of the algorithm increases as the number of dangling nodes increases.

Many other algorithms have been proposed for computing PageRank, including classical iterative methods [1, 4, 30], Krylov subspace methods [13, 14], extrapolation methods [5, 6, 20, 26, 25], and aggregation/disaggregation methods [8, 22, 31]; see also the survey papers [2, 28] and the book [29]. Our algorithm is faster than the power method applied to the full Google matrix $G$, but retains all the advantages of the power method: It is simple to implement and requires minimal storage. Unlike Krylov subspace methods, our algorithm exhibits predictable convergence behavior and is insensitive to changes in the matrix [13]. Moreover, our algorithm should become more competitive as the Web frontier expands and the number of dangling nodes increases. The algorithms in [30, 32] are special cases of our algorithm because our algorithm allows the dangling node and personalization vectors to be different, and thereby facilitates the implementation of TrustRank [18]. TrustRank is designed to diminish the harm done by link spamming and was patented by Google in March 2005 [35]. Moreover, our algorithm can be extended to a more general Google matrix that contains several different dangling node vectors (section 3.4).

In section 4 we examine how the PageRanks of the dangling and nondangling nodes influence each other, as well as the effect of the dangling node vector $w$ on the PageRanks of dangling and nondangling nodes. In particular we show (Theorem 4.1) that the PageRanks of the dangling nodes depend strongly on the PageRanks of the nondangling nodes but not vice versa. Finally, in section 5, we consider a (theoretical) extreme case, where the Web consists solely of dangling nodes. We present a Jordan decomposition for general rank-one matrices (Theorems 5.1 and 5.2) and deduce from it a Jordan decomposition for a Google matrix of rank one (Corollary 5.3).

## 2. The ingredients.

Let $n$ be the number of Web pages and $k$ the number of nondangling nodes among the Web pages, $1 \leq k < n$. We model the link structure of the Web by the $n \times n$ matrix

$$H \equiv \begin{bmatrix} H_{11} & H_{12} \\ 0 & 0 \end{bmatrix},$$

where the $k \times k$ matrix $H_{11}$ represents the links among the nondangling nodes, and $H_{12}$ represents the links from nondangling to dangling nodes; see Figure 2.1. The $n-k$ zero rows in $H$ are associated with the *dangling nodes.*

The elements in the nonzero rows of $H$ are nonnegative and sum to one,

$$H_{11} \geq 0, \qquad H_{12} \geq 0, \qquad H_{11}e + H_{12}e = e, \qquad \text{where} \quad e \equiv \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix},$$

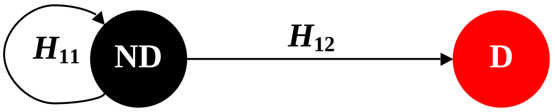


FIG. 2.1. *A simple model of the link structure of the Web. The sphere ND represents the set of nondangling nodes, and D represents the set of dangling nodes. The submatrix $H_{11}$ represents all the links from nondangling nodes to nondangling nodes, while the submatrix $H_{12}$ represents links from nondangling to dangling nodes.*

and the inequalities are to be interpreted elementwise. To obtain a stochastic matrix, we add artificial links to the dangling nodes. That is, we replace each zero row in $H$ by the same *dangling node vector*

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \qquad w \geq 0, \qquad \|w\| \equiv w^T e = 1.$$

Here $w_1$ is $k \times 1$, $w_2$ is $(n-k) \times 1$, $\|\cdot\|$ denotes the one norm (maximal column sum), and the superscript $T$ denotes the transpose. The resulting matrix

$$S \equiv H + dw^T = \begin{bmatrix} H_{11} & H_{12} \\ ew_1^T & ew_2^T \end{bmatrix}, \qquad \text{where} \quad d \equiv \begin{bmatrix} 0 \\ e \end{bmatrix},$$

is stochastic, that is, $S \geq 0$ and $Se = e$.

Finally, so as to work with a stochastic matrix that has a unique stationary distribution, one selects a *personalization vector*

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \qquad v \geq 0, \qquad \|v\| = 1,$$

where $v_1$ is $k \times 1$ and $v_2$ is $(n-k) \times 1$, and defines the Google matrix as the convex combination

$$G \equiv \alpha S + (1-\alpha) ev^T, \qquad 0 \leq \alpha < 1.$$

Although the stochastic matrix $G$ may not be primitive or irreducible, its eigenvalue 1 is distinct and the magnitude of all other eigenvalues is bounded by $\alpha$ [12, 19, 25, 26, 34]. Therefore $G$ has a unique stationary distribution,

$$\pi^T G = \pi^T, \qquad \pi \geq 0, \qquad \|\pi\| = 1.$$

The stationary distribution $\pi$ is called *PageRank*. Element $i$ of $\pi$ represents the PageRank for Web page $i$.

If we partition the PageRank conformally with $G$,

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix},$$

then $\pi_1$ represents the PageRank associated with the nondangling nodes and $\pi_2$ represents the PageRank of the dangling nodes.

The identity matrix of order $n$ will be denoted by $I_n \equiv [e_1 \cdots e_n]$, or simply by $I$.

**3. Lumping.** We show that lumping can be viewed as a similarity transformation of the Google matrix; we derive an expression for PageRank in terms of the stationary distribution of the lumped matrix; we present an algorithm for computing PageRank that is based on lumping; and we extend everything to a Google matrix that has several different dangling node vectors, when it is required to distinguish among different classes of dangling nodes.

It was observed in [32] that the Google matrix represents a lumpable Markov chain. The concept of *lumping* was originally introduced for general Markov matrices, to speed up the computation of the stationary distribution or to obtain bounds [9, 17, 24, 27]. Below we paraphrase lumpability [27, Theorem 6.3.2] in matrix terms: Let $P$ be a permutation matrix and

$$PMP^T = \begin{bmatrix} M_{11} & \cdots & M_{1,k+1} \\ \vdots & & \vdots \\ M_{k+1,1} & \cdots & M_{k+1,k+1} \end{bmatrix}$$

be a partition of a stochastic matrix $M$. Then $M$ is *lumpable* with respect to this partition if each vector $M_{ij}e$ is a multiple of the all-ones vector $e$, $i \neq j$, $1 \leq i, j \leq k+1$.

The Google matrix $G$ is lumpable if all dangling nodes are lumped into a single node [32, Proposition 1]. We condense the notation in section 2 and write the Google matrix as

$$G = \begin{bmatrix} G_{11} & G_{12} \\ eu_1^T & eu_2^T \end{bmatrix}, \qquad \text{where} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \equiv \alpha w + (1-\alpha)v, \tag{3.1}$$

$G_{11}$ is $k \times k$, and $G_{12}$ is $(n-k) \times k$. Here element $(i, j)$ of $G_{11}$ corresponds to block $M_{ij}$, $1 \leq i, j \leq k$; row $i$ of $G_{12}$ corresponds to block $M_{i,k+1}$, $1 \leq i \leq k$; column $i$ of $eu_1^T$ corresponds to $M_{k+1,i}$, $1 \leq i \leq k$; and $eu_2^T$ corresponds to $M_{k+1,k+1}$.

### 3.1. Similarity transformation.

We show that lumping the dangling nodes in the Google matrix can be accomplished by a similarity transformation that reduces $G$ to block upper triangular form.

THEOREM 3.1. *With the notation in section* 2 *and the matrix* $G$ *as partitioned in* (3.1)*, let*

$$X \equiv \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}, \qquad \textit{where} \quad L \equiv I_{n-k} - \frac{1}{n-k}\hat{e}e^T \quad \textit{and} \quad \hat{e} \equiv e - e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

*Then*

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}, \qquad \textit{where} \quad G^{(1)} \equiv \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}.$$

*The matrix* $G^{(1)}$ *is stochastic of order* $k+1$ *with the same nonzero eigenvalues as* $G$.

*Proof.* From

$$X^{-1} = \begin{bmatrix} I_k & 0 \\ 0 & L^{-1} \end{bmatrix}, \qquad L^{-1} = I_{n-k} + \hat{e}e^T,$$

it follows that

$$XGX^{-1} = \begin{bmatrix} G_{11} & G_{12}(I + \hat{e}e^T) \\ e_1 u_1^T & e_1 u_2^T (I + \hat{e}e^T) \end{bmatrix}$$

has the same eigenvalues as $G$. In order to reveal the eigenvalues, we choose a different partitioning and separate the leading $k+1$ rows and columns, and observe that

$$G_{12}(I + \hat{e}e^T)e_1 = G_{12}e, \qquad u_2^T(I + \hat{e}e^T)e_1 = u_2^T e$$

to obtain the block triangular matrix

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}$$

with at least $n-k-1$ zero eigenvalues. □

**3.2. Expression for PageRank.** We give an expression for the PageRank $\pi$ in terms of the stationary distribution $\sigma$ of the small matrix $G^{(1)}$.

Theorem 3.2. *With the notation in section* 2 *and the matrix $G$ as partitioned in* (3.1)*, let*

$$\sigma^T \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix} = \sigma^T, \qquad \sigma \geq 0, \qquad \|\sigma\| = 1$$

*and partition $\sigma^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma_{k+1} \end{bmatrix}$, where $\sigma_{k+1}$ is a scalar. Then the PageRank equals*

$$\pi^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}.$$

*Proof.* As in the proof of Theorem 3.1, we write

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & G^{(2)} \\ 0 & 0 \end{bmatrix},$$

where

$$G^{(2)} \equiv \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} (I + \hat{e}e^T)[e_2 \cdots e_{n-k}].$$

The vector $\begin{bmatrix} \sigma^T & \sigma^T G^{(2)} \end{bmatrix}$ is an eigenvector for $XGX^{-1}$ associated with the eigenvalue $\lambda = 1$. Hence

$$\hat{\pi} \equiv \begin{bmatrix} \sigma^T & \sigma^T G^{(2)} \end{bmatrix} X$$

is an eigenvector of $G$ associated with $\lambda = 1$ and a multiple of the stationary distribution $\pi$ of $G$. Since $G^{(1)}$ has the same nonzero eigenvalues as $G$, and the dominant eigenvalue 1 of $G$ is distinct [12, 19, 25, 26, 34], the stationary distribution $\sigma$ of $G^{(1)}$ is unique.

Next we express $\hat{\pi}$ in terms of quantities in the matrix $G$. We return to the original partitioning which separates the leading $k$ elements,

$$\hat{\pi}^T = \begin{bmatrix} \sigma_{1:k}^T & \begin{pmatrix} \sigma_{k+1} & \sigma^T G^{(2)} \end{pmatrix} \end{bmatrix} \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}.$$

Multiplying out

$$\hat{\pi}^T = \begin{bmatrix} \sigma_{1:k}^T & \begin{pmatrix} \sigma_{k+1} & \sigma^T G^{(2)} \end{pmatrix} L \end{bmatrix}$$

shows that $\hat{\pi}$ has the same leading $k$ elements as $\sigma$.

We now examine the trailing $n-k$ components of $\hat{\pi}^T$. To this end we partition the matrix $L = I_{n-k} - \frac{1}{n-k}\hat{e}e$ and distinguish the first row and column,

$$L = \begin{bmatrix} 1 & 0 \\ -\frac{1}{n-k}e & I - \frac{1}{n-k}ee^T \end{bmatrix}.$$

Then the eigenvector part associated with the dangling nodes is

$$z^T \equiv \begin{bmatrix} \sigma_{k+1} & \sigma^T G^{(2)} \end{bmatrix} L = \begin{bmatrix} \sigma_{k+1} - \frac{1}{n-k}\sigma^T G^{(2)} e & \sigma^T G^{(2)} \left( I - \frac{1}{n-k}ee^T \right) \end{bmatrix}.$$

To remove the terms containing $G^{(2)}$ in $z$, we simplify

$$(I + \hat{e}e^T)[e_2 \cdots e_{n-k}]e = (I + \hat{e}e^T)\hat{e} = (n-k)\hat{e}.$$

Hence

$$G^{(2)} e = (n-k) \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} \hat{e} \tag{3.2}$$

and

$$\begin{aligned} \frac{1}{n-k}\sigma^T G^{(2)} e &= \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} \hat{e} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1 \\ &= \sigma_{k+1} - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1, \end{aligned}$$

where we used $\hat{e} = e - e_1$, and the fact that $\sigma$ is the stationary distribution of $G^{(1)}$, so

$$\sigma_{k+1} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e.$$

Therefore the leading element of $z$ equals

$$z_1 = \sigma_{k+1} - \frac{1}{n-k}\sigma^T G^{(2)} e = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1.$$

For the remaining elements of $z$, we use (3.2) to simplify

$$G^{(2)} \left( I - \frac{1}{n-k}ee^T \right) = G^{(2)} - \frac{1}{n-k}G^{(2)}ee^T = G^{(2)} - \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} \hat{e}e^T.$$

Replacing

$$(I + \hat{e}e^T)[e_2 \cdots e_{n-k}] = [e_2 \cdots e_{n-k}] + \hat{e}e^T$$

in $G^{(2)}$ yields

$$z_{2:n-k}^T = \sigma^T G^{(2)} \left( I - \frac{1}{n-k}ee^T \right) = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} [e_2 \cdots e_{n-k}].$$

Therefore the eigenvector part associated with the dangling nodes is

$$z = \begin{bmatrix} z_1 & z_{2:n-k}^T \end{bmatrix} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}$$

and

$$\hat{\pi} = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}.$$

Since $\pi$ is unique, as discussed in section 2, we conclude that $\hat{\pi} = \pi$ if $\hat{\pi}^T e = 1$. This follows, again, from the fact that $\sigma$ is the stationary distribution of $G^{(1)}$ and $\sigma^T \left[ \begin{smallmatrix} G_{12} \\ u_2^T \end{smallmatrix} \right] e = \sigma_{k+1}$. □

**3.3. Algorithm.** We present an algorithm, based on Theorem 3.2, for computing the PageRank $\pi$ from the stationary distribution $\sigma$ of the lumped matrix

$$G^{(1)} \equiv \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}.$$

The input to the algorithm consists of the nonzero elements of the hyperlink matrix $H$, the personalization vector $v$, the dangling node vector $w$, and the amplification factor $\alpha$. The output of the algorithm is an approximation $\hat{\pi}$ to the PageRank $\pi$, which is computed from an approximation $\hat{\sigma}$ of $\sigma$.

Algorithm 3.1.

```
% Inputs: $H$, $v$, $w$, $\alpha$        Output: $\hat{\pi}$
% Power method applied to $G^{(1)}$:
Choose a starting vector $\hat{\sigma}^T = [\hat{\sigma}_{1:k}^T \quad \hat{\sigma}_{k+1}]$ with $\hat{\sigma} \geq 0$, $\|\hat{\sigma}\| = 1$.
While not converged
    $\hat{\sigma}_{1:k}^T = \alpha\hat{\sigma}_{1:k}^T H_{11} + (1-\alpha)v_1^T + \alpha\hat{\sigma}_{k+1}w_1^T$
    $\hat{\sigma}_{k+1} = 1 - \hat{\sigma}_{1:k}^T e$
end while
% Recover PageRank:
$\hat{\pi}^T = [\hat{\sigma}_{1:k}^T \quad \alpha\hat{\sigma}_{1:k}^T H_{12} + (1-\alpha)v_2^T + \alpha\hat{\sigma}_{k+1}w_2^T]$.
```

Each iteration of the power method applied to $G^{(1)}$ involves a sparse matrix vector multiply with the $k \times k$ matrix $H_{11}$ as well as several vector operations. Thus the dangling nodes are excluded from the power method computation. The convergence rate of the power method applied to $G$ is $\alpha$ [23]. Algorithm 3.1 has the same convergence rate, because $G^{(1)}$ has the same nonzero eigenvalues as $G$ (see Theorem 3.1), but is much faster because it operates on a smaller matrix whose dimension does not depend on the number of dangling nodes. The final step in Algorithm 3.1 recovers $\pi$ via a single sparse matrix vector multiply with the $k \times (n-k)$ matrix $H_{12}$, as well as several vector operations.

Algorithm 3.1 is significantly faster than the power method applied to the full Google matrix $G$, but it retains all advantages of the power method: It is simple to implement and requires minimal storage. Unlike Krylov subspace methods, Algorithm 3.1 exhibits predictable convergence behavior and is insensitive to changes in the matrix [13]. The methods in [30, 32] are special cases of Algorithm 3.1 because they allow the dangling node vector to be different from the personalization vector, thereby facilitating the implementation of TrustRank [18]. TrustRank allows zero elements in the personalization vector $v$ in order to diminish the harm done by link spamming. Algorithm 3.1 can also be extended to the situation when the Google matrix has several different dangling node vectors; see section 3.4.

The power method in Algorithm 3.1 corresponds to Stage 1 of the algorithm in [32]. However, Stage 2 of that algorithm involves the power method on a rank-two matrix of order $n - k + 1$. In contrast, Algorithm 3.1 simply performs a single matrix vector multiply with the $k \times (n-k)$ matrix $H_{12}$. There is no proof that the two-stage algorithm in [32] does compute the PageRank.

**3.4. Several dangling node vectors.** So far we have treated all dangling nodes in the same way, by assigning them the same dangling node vector $w$. However, one dangling node vector may be inadequate for an advanced Web search. For instance, one may want to distinguish different types of dangling node pages based on their functions (e.g., text files, image files, videos, etc.); or one may want to personalize a Web search and assign different vectors to dangling node pages pertaining to different topics, different languages, or different domains; see the discussion in [32, section 8.2].

To facilitate such a model for an advanced Web search, we extend the single class of dangling nodes to $m \geq 1$ different classes, by assigning a different dangling node vector $w_i$ to each class, $1 \leq i \leq m$. As a consequence we need to extend lumping to a more general Google matrix that is obtained by replacing the $n-k$ zero rows in the hyperlink matrix $H$ by $m \geq 1$ possibly different dangling node vectors $w_1, \ldots, w_m$. The more general Google matrix is

$$F \equiv \begin{array}{c} \\ k \\ k_1 \\ \vdots \\ k_m \end{array} \begin{array}{c} \begin{array}{cccc} k & k_1 & \ldots & k_m \end{array} \\ \begin{pmatrix} F_{11} & F_{12} & \cdots & F_{1,m+1} \\ eu_{11}^T & eu_{12}^T & \cdots & eu_{1,m+1}^T \\ \vdots & \vdots & & \vdots \\ eu_{m1}^T & eu_{m2}^T & \cdots & eu_{m,m+1}^T \end{pmatrix} \end{array},$$

where

$$u_i \equiv \begin{bmatrix} u_{i1} \\ \vdots \\ u_{i,m+1} \end{bmatrix} \equiv \alpha w_i + (1-\alpha)v.$$

Let $\tilde{\pi}$ be the PageRank associated with $F$,

$$\tilde{\pi}^T F = \tilde{\pi}^T, \qquad \tilde{\pi} \geq 0, \qquad \|\tilde{\pi}\| = 1.$$

We explain our approach for the case when $F$ has two types of dangling nodes,

$$F = \begin{array}{c} \\ k \\ k_1 \\ k_2 \end{array} \begin{array}{c} \begin{array}{ccc} k & k_1 & k_2 \end{array} \\ \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ eu_{11}^T & eu_{12}^T & eu_{13}^T \\ eu_{21}^T & eu_{22}^T & eu_{23}^T \end{pmatrix} \end{array}.$$

We perform the lumping by a sequence of similarity transformations that starts at the bottom of the matrix. The first similarity transformation lumps the dangling nodes represented by $u_2$ and leaves the leading block of order $k + k_1$ unchanged,

$$X_1 \equiv \begin{array}{c} \\ k+k_1 \\ k_2 \end{array} \begin{array}{c} \begin{array}{cc} k+k_1 & k_2 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & L_1 \end{pmatrix} \end{array},$$

where $L_1$ lumps the $k_2$ trailing rows and columns of $F$,

$$L_1 \equiv I - \frac{1}{k_2}\hat{e}e^T, \quad L_1^{-1} \equiv I + \hat{e}e^T, \qquad \hat{e} = e - e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Applying the similarity transformation to $F$ gives

$$X_1FX_1^{-1}=\begin{array}{cc} & \begin{array}{cccc} k & k_1 & 1 & k_2-1\end{array}\\ \begin{array}{c} k\\ k_1\\ 1\\ k_2-1\end{array} & \begin{pmatrix} F_{11} & F_{12} & F_{13}e & \tilde{F}_{13}\\ eu_{11}^T & eu_{12}^T & (u_{13}^Te)e & e\tilde{u}_{13}^T\\ u_{21}^T & u_{22}^T & u_{23}^Te & \tilde{u}_{23}^T\\ 0 & 0 & 0 & 0 \end{pmatrix}\end{array}$$

with

$$\tilde{F}_{13}\equiv F_{13}L_1^{-1}\begin{bmatrix}e_2 & \cdots & e_{k_2}\end{bmatrix}, \qquad \tilde{u}_{j3}^T\equiv u_{j3}^TL_1^{-1}\begin{bmatrix}e_2 & \cdots & e_{k_2}\end{bmatrix}, \quad j=1,2.$$

The leading diagonal block of order $k+k_1+1$ is a stochastic matrix with the same nonzero eigenvalues as $F$. Before applying the second similarity transformation that lumps the dangling nodes represented by $u_1$, we move the rows with $u_1$ (and corresponding columns) to the bottom of the nonzero matrix, merely to keep the notation simple. The move is accomplished by the permutation matrix

$$P_1\equiv[e_1\cdots e_k \quad e_{k+k_1+1} \quad e_{k+1}\cdots e_{k+k_1} \quad e_{k+k_1+2}\cdots e_n].$$

The symmetrically permuted matrix

$$P_1X_1FX_1^{-1}P_1^T=\left[\begin{array}{ccc|c} F_{11} & F_{13}e & F_{12} & \tilde{F}_{13}\\ u_{21}^T & u_{23}^Te & u_{22}^T & \tilde{u}_{23}^T\\ eu_{11}^T & (u_{13}^Te)e & eu_{12}^T & e\tilde{u}_{13}^T\\ \hline 0 & 0 & 0 & 0 \end{array}\right]$$

retains a leading diagonal block that is stochastic. Now we repeat the lumping on dangling nodes represented by $u_1$, by means of the similarity transformation

$$X_2\equiv\begin{array}{cc} & \begin{array}{ccc} k+1 & k_1 & k_2-1\end{array}\\ \begin{array}{c} k+1\\ k_1\\ k_2-1\end{array} & \begin{pmatrix} I & 0 & 0\\ 0 & L_2 & 0\\ 0 & 0 & I \end{pmatrix}\end{array},$$

where $L_2$ lumps the trailing $k_1$ nonzero rows,

$$L_2\equiv I-\frac{1}{k_1}\hat{e}e^T, \qquad L_2^{-1}\equiv I+\hat{e}e^T.$$

The similarity transformation produces the lumped matrix

$$X_2P_1X_1FX_1^{-1}P_1^TX_2^{-1}=\begin{array}{cc} & \begin{array}{ccccc} k & 1 & 1 & k_1-1 & k_2-1\end{array}\\ \begin{array}{c} k\\ 1\\ 1\\ k_1-1\\ k_2-1\end{array} & \begin{pmatrix} F_{11} & F_{13}e & F_{12}e & \tilde{F}_{12} & \tilde{F}_{13}\\ u_{21}^T & u_{23}^Te & u_{22}^Te & \tilde{u}_{22}^T & \tilde{u}_{23}^T\\ u_{11}^T & u_{13}^Te & u_{12}^Te & \tilde{u}_{12}^T & \tilde{u}_{13}^T\\ 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}\end{array}.$$

Finally, for notational purposes, we restore the original ordering of dangling nodes by permuting rows and columns $k+1$ and $k+2$,

$$P_2\equiv[e_1\cdots e_k \quad e_{k+2} \quad e_{k+1} \quad e_{k+3}\cdots e_n].$$

The final lumped matrix is

$$P_2X_2P_1X_1FX_1^{-1}P_1^TX_2^{-1}P_2^T = \begin{bmatrix} F_{11} & F_{12}e & F_{13}e & * \\ u_{11}^T & u_{12}^Te & u_{13}^Te & * \\ u_{21}^Te & u_{22}^Te & u_{23}^Te & * \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} F^{(1)} & * \\ 0 & 0 \end{bmatrix}.$$

The above discussion for $m = 2$ illustrates how to extend Theorems 3.1 and 3.2 to any number $m$ of dangling node vectors.

THEOREM 3.3. *Define* $X_i$ *as*

$$\begin{array}{ll} & \begin{array}{ccc} k+(i-1)+\sum_{j=1}^{m-i}k_j & k_{m-i+1} & 1-i+\sum_{j=m-i+2}^{m}k_j \end{array} \\ \begin{array}{l} k+(i-1)+\sum_{j=1}^{m-i}k_j \\ k_{m-i+1} \\ 1-i+\sum_{j=m-i+2}^{m}k_j \end{array} & \left(\begin{array}{ccc} I & 0 & 0 \\ 0 & L_i & 0 \\ 0 & 0 & I \end{array}\right) \end{array}$$

*and*

$$P_i \equiv [e_1 \cdots e_k \quad e_{r+i} \quad e_{k+1} \cdots e_{r+i-1} \quad e_{r+i+1} \cdots e_n], \qquad r = k + \sum_{j=1}^{m-i} k_j.$$

*Then*

$$P_mX_mP_{m-1}X_{m-1} \cdots P_1X_1FX_1^{-1}P_1^T \cdots X_m^{-1}P_m^T = \begin{bmatrix} F^{(1)} & * \\ 0 & 0 \end{bmatrix},$$

*where the lumped matrix*

$$F^{(1)} \equiv \begin{bmatrix} F_{11} & F_{12}e & \cdots & F_{1,m+1}e \\ u_{11}^T & u_{12}^Te & \cdots & u_{1,m+1}^Te \\ \vdots & \vdots & & \vdots \\ u_{m1}^T & u_{m2}^Te & \cdots & u_{m,m+1}^Te \end{bmatrix}$$

*is stochastic of order* $k+m$ *with the same nonzero eigenvalues as* $F$.

THEOREM 3.4. *Let* $\rho$ *be the stationary distribution of the lumped matrix*

$$F^{(1)} \equiv \begin{bmatrix} F_{11} & F_{12}e & \cdots & F_{1,m+1}e \\ u_{11}^T & u_{12}^Te & \cdots & u_{1,m+1}^Te \\ \vdots & \vdots & & \vdots \\ u_{m1}^T & u_{m2}^Te & \cdots & u_{m,m+1}^Te \end{bmatrix}; \tag{3.3}$$

*that is,*

$$\rho^TF^{(1)} = \rho^T, \qquad \rho \geq 0, \qquad \|\rho\| = 1.$$

*With the partition* $\rho^T = \begin{bmatrix} \rho_{1:k}^T & \rho_{k+1:k+m}^T \end{bmatrix}$, *where* $\rho_{k+1:k+m}$ *is* $m \times 1$, *the PageRank of* $F$ *equals*

$$\tilde{\pi}^T = \left[ \rho_{1:k}^T \quad \rho^T \begin{pmatrix} F_{12} & \cdots & F_{1,m+1} \\ u_{12}^T & \cdots & u_{1,m+1}^T \\ \vdots & & \vdots \\ u_{m2}^T & \cdots & u_{m,m+1}^T \end{pmatrix} \right].$$

**4. PageRanks of dangling versus nondangling nodes.** We examine how the PageRanks of dangling and nondangling nodes influence each other, as well as the effect of the dangling node vector on the PageRanks.

From Theorem 3.2 and Algorithm 3.1, we see that the PageRank $\pi_1$ of the nondangling nodes can be computed separately from the PageRank $\pi_2$ of the dangling nodes, and that $\pi_2$ depends directly on $\pi_1$. The expressions below make this even clearer.

Theorem 4.1. *With the notation in section* 2*,*

$$\begin{aligned}\pi_1^T &= \left((1-\alpha)v_1^T + \rho w_1^T\right)(I-\alpha H_{11})^{-1},\\ \pi_2^T &= \alpha\pi_1^T H_{12} + (1-\alpha)v_2^T + \alpha(1-\|\pi_1\|)w_2^T,\end{aligned}$$

*where*

$$\rho \equiv \alpha\frac{1-(1-\alpha)v_1^T(I-\alpha H_{11})^{-1}e}{1+\alpha w_1^T(I-\alpha H_{11})^{-1}e} \geq 0.$$

*Proof.* Rather than using Theorem 3.2 we found it easier just to start from scratch. From $G = \alpha(H+dw^T)+(1-\alpha)v^T$ and the fact that $\pi^T e = 1$, it follows that $\pi$ is the solution to the linear system

$$\pi^T = (1-\alpha)v^T\left(I-\alpha H-\alpha dw^T\right)^{-1},$$

whose coefficient matrix is a strictly row diagonally dominant M-matrix [1, equation (5)], [4, equation (2), Proposition 2.4]. Since $R \equiv I-\alpha H$ is also an M-matrix, it is nonsingular, and the elements of $R^{-1}$ are nonnegative [3, section 6]. The Sherman–Morrison formula [15, section 2.1.3] implies that

$$\left(R-\alpha dw^T\right)^{-1} = R^{-1} + \frac{\alpha R^{-1}dw^TR^{-1}}{1-\alpha w^TR^{-1}d}.$$

Substituting this into the expression for $\pi$ gives

$$\pi^T = (1-\alpha)v^TR^{-1} + \frac{\alpha(1-\alpha)v^TR^{-1}d}{1-\alpha w^TR^{-1}d}w^TR^{-1}. \tag{4.1}$$

We now show that the denominator $1-\alpha w^TR^{-1}d > 0$. Using the partition

$$R^{-1} = (I-\alpha H)^{-1} = \begin{bmatrix}(I-\alpha H_{11})^{-1} & \alpha(I-\alpha H_{11})^{-1}H_{12}\\ 0 & I\end{bmatrix}$$

yields

$$1-\alpha w^TR^{-1}d = 1-\alpha\left(\alpha w_1^T(I-\alpha H_{11})^{-1}H_{12}e + w_2^Te\right). \tag{4.2}$$

Rewrite the term involving $H_{12}$ by observing that $H_{11}e+H_{12}e = e$ and that $I-\alpha H_{11}$ is an M-matrix, so

$$0 \leq \alpha(I-\alpha H_{11})^{-1}H_{12}e = e-(1-\alpha)(I-\alpha H_{11})^{-1}e. \tag{4.3}$$

Substituting this into (4.2) and using $1 = w^Te = w_1^Te + w_2^Te$ shows that the denominator in the Sherman–Morrison formula is positive,

$$1-\alpha w^TR^{-1}d = (1-\alpha)\left(1+\alpha w_1^T(I-\alpha H_{11})^{-1}e\right) > 0.$$

Furthermore, $0 \le \alpha < 1$ implies $1 - \alpha w^T R^{-1} d > 1 - \alpha$.

Substituting the simplified denominator into the expression (4.1) for $\pi$ yields

$$\pi^T = (1-\alpha) v^T R^{-1} + \alpha \frac{v^T R^{-1} d}{1 + \alpha w_1^T (I - \alpha H_{11})^{-1} e} w^T R^{-1}. \tag{4.4}$$

We obtain for $\pi_1$

$$\pi_1^T = \left( (1-\alpha) v_1^T + \alpha \frac{v^T R^{-1} d}{1 + \alpha w_1^T (I - \alpha H_{11})^{-1} e} w_1^T \right) (I - \alpha H_{11})^{-1}.$$

Combining the partitioning of $R^{-1}$, (4.3), and $v_1^T e + v_2^T e = 1$ gives

$$\begin{aligned} 0 \le v^T R^{-1} d &= \begin{bmatrix} v_1^T & v_2^T \end{bmatrix} \begin{bmatrix} (I - \alpha H_{11})^{-1} & \alpha (I - \alpha H_{11})^{-1} H_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ e \end{bmatrix} \\ &= \alpha v_1^T (I - \alpha H_{11})^{-1} H_{12} e + v_2^T e \\ &= 1 - (1-\alpha) v_1^T (I - \alpha H_{11})^{-1} e. \end{aligned}$$

Hence $\pi_1^T = \left( (1-\alpha) v_1^T + \rho w_1^T \right) (I - \alpha H_{11})^{-1}$ with $\rho > 0$.

To obtain the expression for $\pi_2$, observe that the second block element in

$$\pi^T (I - \alpha H - \alpha d w^T) = (1-\alpha) v^T$$

equals

$$-\alpha \pi_1^T H_{12} + \pi_2^T - \alpha \pi_2^T e w_2^T = (1-\alpha) v_2^T.$$

The result follows from $\pi_1^T e + \pi_2^T e = 1$. $\square$

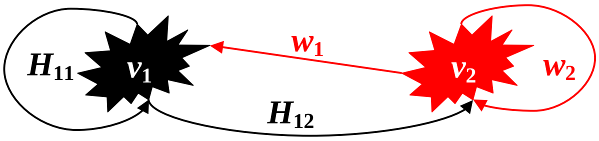


Fig. 4.1. *Sources of PageRank. Nondangling nodes receive their PageRank from $v_1$ and $w_1$, distributed through the links $H_{11}$. In contrast, the PageRank of the dangling nodes comes from $v_2$, $w_2$, and the PageRank of the nondangling nodes through the links $H_{12}$.*

*Remark* 4.1. We draw the following conclusions from Theorem 4.1 with regard to how dangling and nondangling nodes accumulate PageRank; see Figure 4.1.

- The PageRank $\pi_1$ of the nondangling nodes does *not* depend on the connectivity among the dangling nodes (elements of $w_2$), the personalization vector for the dangling nodes (elements of $v_2$), or the links from nondangling to dangling nodes (elements of $H_{12}$).
  To be specific, $\pi_1$ does not depend on individual elements of $w_2$, $v_2$, and $H_{12}$. Rather, the dependence is on the norms, through $\|v_2\| = 1 - \|v_1\|$, $\|w_2\| = 1 - \|w_1\|$, and $H_{12} e = e - H_{11} e$.
- The PageRank $\pi_1$ of the nondangling nodes does not depend on the PageRank $\pi_2$ of the dangling nodes or their number, because $\pi_1$ can be computed without knowledge of $\pi_2$.

- The nondangling nodes receive their PageRank $\pi_1$ from their personalization vector $v_1$ and the dangling node vector $w_1$, both of which are distributed through the links $H_{11}$.
- The dangling nodes receive their PageRank $\pi_2$ from three sources: the associated part $v_2$ of the personalization vector; the associated part $w_2$ of the dangling node vector; and the PageRank $\pi_1$ of the nondangling nodes filtered through the connecting links $H_{12}$.

  The links $H_{12}$ determine how much PageRank flows from nondangling to dangling nodes.
- The influence of the associated dangling node vector $w_2$ on the PageRank $\pi_2$ of the dangling nodes diminishes as the combined PageRank $\|\pi_1\|$ of the nondangling nodes increases.

Taking norms in Theorem 4.1 gives a bound on the combined PageRank of the nondangling nodes. As in section 2, the norm is $\|z\| \equiv z^T e$ for $z \geq 0$.

Corollary 4.2. *With the assumptions of Theorem* 4.1,

$$\|\pi_1\| = \frac{(1-\alpha)\|v_1\|_H + \alpha\|w_1\|_H}{1+\alpha\|w_1\|_H},$$

*where* $\|z\|_H \equiv z^T(I-\alpha H_{11})^{-1}e$ *for any* $z \geq 0$ *and*

$$(1-\alpha)\|z\| \leq \|z\|_H \leq \frac{1}{1-\alpha}\|z\|.$$

*Proof.* Since $(I-\alpha H_{11})^{-1}$ is nonsingular with nonnegative elements, $\|\cdot\|_H$ is a norm. Let $\|\cdot\|_\infty$ be the infinity norm (maximal row sum). Then the Hölder inequality [15, section 2.2.2] implies for any $z \geq 0$,

$$\|z\|_H \leq \|z\| \, \|(I-\alpha H_{11})^{-1}\|_\infty \leq \frac{1}{1-\alpha}\|z\|.$$

As for the lower bound,

$$\|z\|_H \geq \|z\| - \alpha z^T H_{11} e \geq (1-\alpha)\|z\|. \qquad \square$$

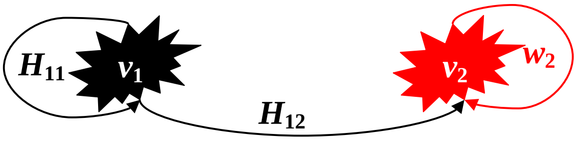


Fig. 4.2. *Sources of PageRank when* $w_1 = 0$. *The nondangling nodes receive their PageRank only from* $v_1$. *The dangling nodes, in contrast, receive their PageRank from* $v_2$ *and* $w_2$, *as well as from the PageRank of the nondangling nodes filtered through the links* $H_{12}$.

Corollary 4.2 implies that the combined PageRank $\|\pi_1\|$ of the nondangling nodes is an increasing function of $\|w_1\|$. In particular, when $w_1 = 0$, the combined PageRank $\|\pi_1\|$ is minimal among all $w$ and the dangling vector $w_2$ has a stronger influence on the PageRank $\pi_2$ of the dangling nodes. The dangling nodes act like a sink and absorb more PageRank because there are no links back to the nondangling nodes; see Figure 4.2. When $w_1 = 0$ we get

$$\begin{aligned} \pi_1^T &= (1-\alpha)v_1^T(I-\alpha H_{11})^{-1}, \\ \pi_2^T &= \alpha\pi_1^T H_{12} + (1-\alpha)v_2^T + \alpha(1-\|\pi_1\|)w_2^T. \end{aligned} \tag{4.5}$$

In the other extreme case when $w_2 = 0$, the dangling nodes are not connected to each other; see Figure 4.3:

$$\begin{aligned} \pi_1^T &= \left((1-\alpha)v_1^T + \rho w_1^T\right)(I - \alpha H_{11})^{-1}, \\ \pi_2^T &= \alpha \pi_1^T H_{12} + (1-\alpha) v_2^T. \end{aligned} \tag{4.6}$$

In this case the PageRank $\pi_1$ of the nondangling nodes has only a positive influence on the PageRank of the dangling nodes.

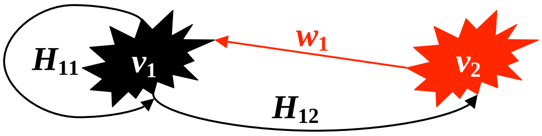


FIG. 4.3. *Sources of PageRank when $w_2 = 0$. The dangling nodes receive their PageRank only from $v_2$, and from the PageRank of the nondangling nodes filtered through the links $H_{12}$.*

An expression for $\pi$ when dangling node and personalization vectors are the same, i.e., $w = v$, was given in [10],

$$\pi^T = (1-\alpha)\left(1 + \frac{\alpha v^T R^{-1} d}{1 - \alpha v^T R^{-1} d}\right) v^T R^{-1}, \qquad \text{where} \quad R \equiv I - \alpha H.$$

In this case the PageRank vector $\pi$ is a multiple of the vector $v^T(I - \alpha H)^{-1}$.

## 5. Only dangling nodes.

We examine the (theoretical) extreme case when all Web pages are dangling nodes. In this case the matrices $S$ and $G$ have rank one. We first derive a Jordan decomposition for general matrices of rank one, before we present a Jordan form for a Google matrix of rank one.

We start with rank-one matrices that are diagonalizable. The vector $e_j$ denotes the $j$th column of the identity matrix $I$.

THEOREM 5.1 (eigenvalue decomposition). *Let $A = yz^T \neq 0$ be a real square matrix with $\lambda \equiv z^T y \neq 0$. If $z$ has an element $z_j \neq 0$, then $X^{-1}AX = \lambda\, e_j e_j^T$, where*

$$X \equiv I + y e_j^T - \frac{1}{z_j} e_j z^T, \qquad X^{-1} = I - e_j e_j^T - \frac{1}{\lambda} y z^T + \frac{1 + y_j}{\lambda} e_j z^T.$$

*Proof.* The matrix $A$ has a repeated eigenvalue zero and a distinct nonzero eigenvalue $\lambda$ with right eigenvector $y$ and left eigenvector $z$. From $\lambda y e_j^T = AX = \lambda X e_j e_j^T$ and $X^{-1}A = e_j z^T$ it follows that $X^{-1}X = I$ and $X^{-1}AX = \lambda e_j e_j^T$. □

Now we consider rank-one matrices that are not diagonalizable. In this case all eigenvalues are zero, and the matrix has a Jordan block of order two.

THEOREM 5.2 (Jordan decomposition). *Let $A = yz^T \neq 0$ be a real square matrix with $z^T y = 0$. Then $y$ and $z$ have elements $y_j z_j \neq 0 \neq y_k z_k$, $j < k$. Define a symmetric permutation matrix $P$ so that $Pe_k = e_{j+1}$ and $Pe_j = e_j$. Set $\hat{y} \equiv Py$ and $\hat{u} \equiv Pz - e_{j+1}$. Then $X^{-1}AX = e_j e_{j+1}^T$ with*

$$X \equiv P\left(I + \hat{y} e_j^T - \frac{1}{\hat{u}_j} e_j \hat{u}^T\right), \qquad X^{-1} = \left(I - e_j e_j^T + \frac{1}{\hat{y}_k} \hat{y} \hat{u}^T - \frac{1 + \hat{y}_j}{\hat{y}_k} e_j \hat{u}^T\right) P.$$

*Proof.* To satisfy $z^T y = 0$ for $y \neq 0$ and $z \neq 0$, we must have $y_j z_j \neq 0$ and $y_k z_k \neq 0$ for some $j < k$.

Since $A$ is a rank-one matrix with all eigenvalues equal to zero, it must have a Jordan block of the form $\left[\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right]$. To reveal this Jordan block, set $\hat{z} \equiv Pz$,

$$\hat{X} \equiv \left(I + \hat{y}e_j^T - \frac{1}{\hat{u}_j}e_j\hat{u}^T\right), \qquad \hat{X}^{-1} = \left(I - e_je_j^T + \frac{1}{\hat{y}_k}\hat{y}\hat{u}^T - \frac{1+\hat{y}_j}{\hat{y}_k}e_j\hat{u}^T\right).$$

Then the matrix $\hat{A} \equiv \hat{y}\hat{z}^T$ has a Jordan decomposition $\hat{X}^{-1}\hat{A}\hat{X} = e_je_{j+1}^T$. This follows from $u_j = z_j$, $\hat{y}e_{j+1}^T = \hat{A}\hat{X} = \hat{X}e_je_{j+1}^T$, and $\hat{X}^{-1}\hat{A} = e_j\hat{z}^T$.

Finally, we undo the permutation by means of $X \equiv P\hat{X}$, $X^{-1} = \hat{X}^{-1}P$, so that $X^{-1}X = I$ and $X^{-1}AX = e_je_{j+1}^T$. ☐

Theorems 5.1 and 5.2 can also be derived from [21, Theorem 1.4].

In the (theoretical) extreme case when all Web pages are dangling nodes, the Google matrix is diagonalizable of rank one.

Corollary 5.3 (rank-one Google matrix). *With the notation in section* 2 *and* (3.1), *let* $G = eu^T$. *Let* $u_j \neq 0$ *be a nonzero element of* $u$. *Then* $X^{-1}GX = e_je_j^T$ *with*

$$X = I + ee_j^T - \frac{1}{v_j}e_ju^T$$

*and*

$$X^{-1} = I - e_je_j^T - eu^T + 2e_ju^T.$$

*In particular,* $\pi^T = e_j^TX^{-1} = u^T$.

*Proof.* Since $1 = u^Te \neq 0$, the Google matrix is diagonalizable, and the expression in Theorem 5.1 applies. ☐

Corollary 5.3 can also be derived from [34, Theorems 2.1, 2.3].

## REFERENCES

[1] A. Arasu, J. Novak, A. Tomkins, and J. Tomlin, *PageRank computation and the structure of the web: Experiments and algorithms*, in Proceedings of the Eleventh International World Wide Web Conference (WWW2002), ACM Press, New York, 2002. Available online at http://www2002.org/CDROM/poster/173.pdf.

[2] P. Berkhin, *A survey on PageRank computing*, Internet Math., 2 (2005), pp. 73–120.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[4] M. Bianchini, M. Gori, and F. Scarselli, *Inside PageRank*, ACM Transactions on Internet Technology, 5 (2005), pp. 92–128.

[5] C. Brezinski and M. Redivo-Zaglia, *The PageRank vector: Properties, computation, approximation, and acceleration*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 551–575.

[6] C. Brezinski, M. Redivo-Zaglia, and S. Serra-Capizzano, *Extrapolation methods for PageRank computations*, C. R. Math. Acad. Sci. Paris, 340 (2005), pp. 393–397.

[7] S. Brin and L. Page, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks and ISDN Systems, 30 (1998), pp. 107–117.

[8] A. Z. Broder, R. Lempel, F. Maghoul, and J. Pedersen, *Efficient PageRank approximation via graph aggregation*, in Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), ACM Press, New York, 2004, pp. 484–485.

[9] T. Dayar and W. J. Stewart, *Quasi lumpability, lower-bounding coupling matrices, and nearly completely decomposable Markov chains*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 482–498.

[10] G. M. Del Corso, A. Gullí, and F. Romani, *Fast PageRank computation via a sparse linear system*, Internet Math., 2 (2005), pp. 251–273. Available online at http://www.internetmathematics.org/volumes/2/3/DelCorso.pdf.

[11] N. Eiron, K. S. McCurley, and J. A. Tomlin, *Ranking the web frontier*, in Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), ACM Press, New York, 2004, pp. 309–318.

[12] L. Eldén, *The Eigenvalues of the Google Matrix*, Technical report LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, Linköping, Sweden, 2004.

[13] D. Gleich, L. Zhukov, and P. Berkhin, *Fast parallel PageRank: A Linear System Approach*, Technical report, Yahoo!, Sunnyvale, CA, 2004.

[14] G. H. Golub and C. Greif, *An Arnoldi-type algorithm for computing PageRank*, BIT, 46 (2006), pp. 759–771.

[15] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[16] A. Gulli and A. Signorini, *The indexable web is more than* 11.5 *billion pages*, in Proceedings of the Fourteenth International World Wide Web Conference (WWW2005), ACM Press, New York, 2005, pp. 902–903.

[17] L. Gurvits and J. Ledoux, *Markov property for a function of a Markov chain: A linear algebra approach*, Linear Algebra Appl., 404 (2005), pp. 85–117.

[18] Z. Gyöngyi, H. Garcia-Molina, and P. J., *Combating web spam with TrustRank*, in Proceedings of the Thirtieth VLDB Conference, ACM Press, New York, 2004, pp. 576–587.

[19] T. H. Haveliwala and S. D. Kamvar, *The Second Eigenvalue of the Google Matrix*, Technical report, Computer Science Department, Stanford University, Palo Alto, CA, 2003.

[20] T. H. Haveliwala, S. D. Kamvar, D. Klein, C. D. Manning, and G. H. Golub, *Computing PageRank Using Power Extrapolation*, Technical report 2003-45, Stanford University, Palo Alto, CA, 2003. Available online at http://dbpubs.stanford.edu/pub/2003-45.

[21] R. A. Horn and S. Serra-Capizzano, *A general setting for the parametric Google matrix*, Internet Math., to appear.

[22] I. C. F. Ipsen and S. Kirkland, *Convergence analysis of a PageRank updating algorithm by Langville and Meyer*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 952–967.

[23] I. C. F. Ipsen and R. S. Wills, *Mathematical properties and analysis of Google's PageRank*, Bol. Soc. Esp. Mat. Apl., 34 (2006), pp. 191–196.

[24] R. W. Jernigan and R. H. Baran, *Testing lumpability in Markov chains*, Statist. Probab. Lett., 64 (2003), pp. 17–23.

[25] S. D. Kamvar, T. H. Haveliwala, and G. H. Golub, *Adaptive methods for the computation of PageRank*, Linear Algebra Appl., 386 (2004), pp. 51–65.

[26] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, in Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Toronto, ACM Press, New York, 2003, pp. 261–270.

[27] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Van Nostrand Co., Princeton, NJ, 1960.

[28] A. N. Langville and C. D. Meyer, *Deeper inside PageRank*, Internet Math., 1 (2004), pp. 335–380. Available online at http://www.internetmathematics.org/volumes/1/3/Langville.pdf.

[29] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.

[30] A. N. Langville and C. D. Meyer, *A reordering for the PageRank problem*, SIAM J. Sci. Comput., 27 (2006), pp. 2112–2120.

[31] A. N. Langville and C. D. Meyer, *Updating Markov chains with an eye on Google's PageRank*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 968–987.

[32] C. P. Lee, G. H. Golub, and S. A. Zenios, *A Fast Two-Stage Algorithm for Computing PageRank and Its Extensions*, Technical report, Stanford University, Palo Alto, CA, 2003.

[33] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999. Available online at http://dbpubs.stanford.edu/pub/1999-66.

[34] S. Serra-Capizzano, *Jordan canonical form of the Google matrix: A potential contribution to the PageRank computation*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 305–312.

[35] R. S. Wills, *Google's PageRank: The math behind the search engine*, Math. Intelligencer, 28 (2006), pp. 6–11.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1297–1321

# MAXIMUM ATTAINABLE ACCURACY OF INEXACT SADDLE POINT SOLVERS*

PAVEL JIRÁNEK[†] AND MIROSLAV ROZLOŽNÍK[‡]

**Abstract.** In this paper we study numerical behavior of several iterative Krylov subspace solvers applied to the solution of large-scale saddle point problems. Two main representatives of segregated solution approach are analyzed: the Schur complement reduction method based on the elimination of primary unknowns and the null-space projection method, which relies on a basis for the subspace described by the constraints. We show that the choice of the back-substitution formula may considerably influence the maximum attainable accuracy of approximate solutions computed in finite precision arithmetic.

**Key words.** saddle point problems, Schur complement reduction method, null-space projection method, rounding error analysis

**AMS subject classifications.** 65F10, 65F20, 65F35

**DOI.** 10.1137/060659727

## 1. Introduction.

We want to solve a saddle point system which is in fact the symmetric indefinite system with $2 \times 2$ block structure

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{1.1}$$

where the diagonal $n \times n$ block $A$ is symmetric positive definite and the $n \times m$ off-diagonal block $B$ has full column rank. Saddle point problems have recently attracted a lot of attention and appear to be a time-critical component in the solution of large-scale problems in many applications of computational science and engineering. A large amount of work has been devoted to a wide selection of solution techniques varying from the fully direct approach, through the use of iterative stationary or Krylov subspace methods, up to the combination of direct and iterative techniques including preconditioned iterative schemes. For an excellent survey on applications, methods, and results on numerical solution of saddle point problems, we refer to [5] and numerous references therein (relevant references will be given later in the text). Significantly less attention, however, has been paid so far to the numerical stability aspects. In this paper we concentrate on the numerical behavior of schemes which compute separately the unknown vectors $x$ and $y$: one of them is first obtained from a reduced system of a smaller dimension and, once it has been computed, the other unknown is obtained by back-substitution solving exactly or inexactly another reduced problem. The main representatives of such a segregated approach are the Schur

---

[*] Received by the editors May 12, 2006; accepted for publication (in revised form) by M. Benzi May 18, 2007; published electronically January 4, 2008.
http://www.siam.org/journals/simax/29-4/65972.html

[†] Department of Modelings of Processes, Technical University of Liberec, Hálkova 6, CZ-461 17 Liberec, Czech Republic (pavel.jiranek@tul.cz). The work of this author was supported by the MSMT CR under the project 1M0554 “Advanced Remedial Technologies.”

[‡] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, CZ-182 07 Prague 8, Czech Republic (miro@cs.cas.cz). The work of this author was supported by the project 1ET400300415 within the National Program of Research “Information Society” and by the Institutional Research Plan AV0Z10300504 “Computer Science for the Information Society: Models, Algorithms, Applications.”

complement reduction method and the null-space projection method. In this paper we analyze such algorithms which can be interpreted as iterations for the reduced system but compute the approximate solutions $x_k$ and $y_k$ to both unknown vectors $x$ and $y$ simultaneously.

The Schur complement reduction method uses the block factorization in the form

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} = \begin{pmatrix} I & 0 \\ B^T A^{-1} & I \end{pmatrix} \begin{pmatrix} A & B \\ 0 & -B^T A^{-1} B \end{pmatrix},$$

where the matrix $-B^T A^{-1} B$ is the Schur complement of $A$ in (1.1). Such decomposition leads to solving the resulting block triangular system

$$\begin{pmatrix} A & B \\ 0 & -B^T A^{-1} B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ -B^T A^{-1} f \end{pmatrix}, \tag{1.2}$$

which is nothing but a block Gaussian elimination applied to the original system (1.1). The block triangular system (1.2) is solved by computing the unknown $y$ from the symmetric positive definite Schur complement system of order $m$ and then by computing the unknown $x$ from a system of order $n$ with the symmetric positive definite matrix $A$. This approach leads to the explicit formula for the unknown vector $x = A^{-1}(f - By)$. The system (1.1) can be seen as two block equations and we refer to them as the "first block equation in (1.1)" and the "second block equation in (1.1)." The null-space projection method is based on the projection of the first block equation in (1.1) onto the null-space $N(B^T)$ and onto its orthogonal complement $R(B)$, respectively. According to the second block equation of (1.1) the unknown $x$ belongs to $N(B^T)$ and therefore we get the block triangular system

$$\begin{pmatrix} (I - \Pi)A(I - \Pi) & 0 \\ B^T A & B^T B \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} (I - \Pi)f \\ B^T f \end{pmatrix}, \tag{1.3}$$

where $\Pi \equiv B(B^T B)^{-1} B^T$ denotes the orthogonal projector onto $R(B)$. This triangular system is solved by back substitution, where we first compute the unknown $x$ from the projected system of order $n$ with the symmetric positive semidefinite matrix $(I - \Pi)A(I - \Pi)$. Once it has been computed, the unknown $y$ is obtained as $y = B^\dagger(f - Ax)$ by solving the least squares problem

$$\|f - Ax - By\| = \min_{v \in \mathbb{R}^m} \|f - Ax - Bv\|, \tag{1.4}$$

where $B^\dagger$ denotes the Moore–Penrose pseudoinverse of $B$. The success of algorithms for solving the block triangular system (1.2) or (1.3) depends on the availability of good approximations to the inverse of the block $A$ or to the pseudoinverse of $B$, respectively. More precisely, one looks for a cheap approximate solution to the inner systems with the matrix $A$ and/or to the associated least squares problems with the matrix $B$. Numerous inexact schemes have been used and analyzed (see, e.g., the analysis of inexact Uzawa algorithms [15, 11, 12, 4, 37], inexact null-space methods [28, 35, 36], multilevel or multigrid methods [10, 9, 36], domain decomposition methods [8], two-stage iterative processes [27, 16], and inner-outer iterations [19]). These works contain mainly the analysis of a convergence delay caused by the inexact solution of inner systems or least squares problems.

In this paper we concentrate on the question of what is the best accuracy we can get from inexact schemes solving either (1.2) or (1.3) when implemented in finite precision arithmetic. The fact that the inner solution tolerance strongly influences the accuracy of computed iterates is known and was studied in several contexts. The general framework for understanding inexact Krylov subspace methods has been developed in [31] and [33]. Assuming exact arithmetic, Simoncini and Szyld [31] and van den Eshof and Sleijpen [33] investigated the effect of an approximately computed matrix-vector product in every iteration on the ultimate accuracy of several solvers and explained the success of relaxation strategies for the inner accuracy tolerance from [7, 8, 18]. The developed theory strongly exploits the particular properties of an iterative method used for solving the associated system. In the context of saddle point problems, this requires a deep analysis of the outer iteration scheme for solving the reduced Schur complement or projected system (in particular, we refer to [31, section 8]).

The effects of rounding errors in the Schur complement reduction method and the null-space projection method have been studied, e.g., in [1, 2, 14, 26], where the maximum attainable accuracy of computed approximate solutions by means of residuals and errors is estimated depending on the user tolerance specified in the outer iteration. In this paper we analyze the influence of the inexact solution of inner systems/least squares problems on the same quantities. Our approach is based on a standard backward analysis which allows us to take into account both the inexactness of the inner iteration loops as well as the accompanying rounding errors that occur in finite precision arithmetic.

The theory developed for the outer iteration process is similar to the analysis of Greenbaum in [22, 21] who estimated the gap between the true and recursively updated residual for a general class of iterative methods using coupled two-term recursions. The difference here is that every computed approximate solution of an inner problem is interpreted as an exact solution of a perturbed problem induced by the actual stopping criterion, while the theory of [22] considered only the rounding errors associated with a fixed matrix-vector multiplication. In contrast to the theory of inexact Krylov methods [31, 33], the bounds for the true residual in the outer iteration loop are obtained without specifying the solver used for solving the Schur complement or the projected Hessian system. It appears that the maximum attainable accuracy level in the outer process is mainly given by the inexactness of solving the inner problems and it is not further magnified by the associated rounding errors. These results are thus similar to ones which can be obtained in exact arithmetic.

The situation is different when looking at the numerical behavior of residuals associated with the original saddle point system, which describe how accurately the two block equations in (1.1) are satisfied. It is shown that the attainable accuracy of computed approximate solutions then depends significantly on the back-substitution formula used for computing the remaining unknowns. Our results show that, independent of the fact that the inner systems are solved inexactly, some back-substitution schemes lead ultimately to residuals on the roundoff unit level. Indeed, our results confirm that, depending on which back-substitution formula is used, the computed iterates may satisfy either the first or the second block equation to the working accuracy. We believe that such results cannot be obtained using the exact arithmetic considerations and are of importance in applications requiring accurate approximations (see, e.g., [20, 17, 13]). On the other hand, we agree that in many applications the saddle point system comes from a discretization of certain partial differential equa-

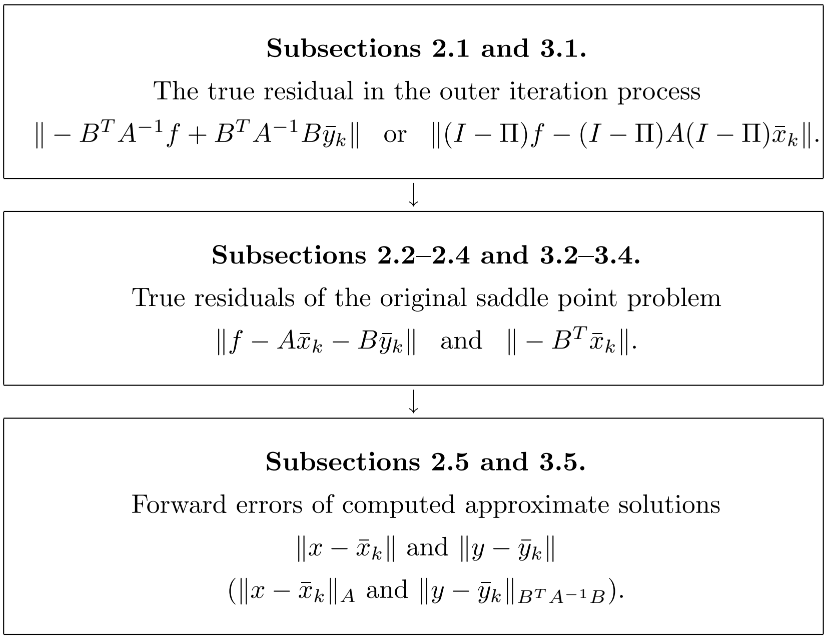

**Subsections 2.1 and 3.1.**
The true residual in the outer iteration process
$\| - B^T A^{-1} f + B^T A^{-1} B \bar{y}_k \|$ or $\|(I - \Pi) f - (I - \Pi) A (I - \Pi) \bar{x}_k \|$.
**Subsections 2.2–2.4 and 3.2–3.4.**
True residuals of the original saddle point problem
$\| f - A\bar{x}_k - B\bar{y}_k \|$ and $\| - B^T \bar{x}_k \|$.
**Subsections 2.5 and 3.5.**
Forward errors of computed approximate solutions
$\| x - \bar{x}_k \|$ and $\| y - \bar{y}_k \|$
($\| x - \bar{x}_k \|_A$ and $\| y - \bar{y}_k \|_{B^T A^{-1} B}$).

FIG. 1.1.

tions and much lower accuracy is sufficient. In any case, our paper gives a theoretical explanation for the behavior which was probably observed or is already implicitly known. However, we have not found any explicit references to this issue. The implementations that we point out as optimal are actually those which are widely used and suggested in applications.

The organization of the paper is as follows. Sections 2 and 3 are devoted to the rounding error analysis of the Schur complement reduction method and the null-space projection method, respectively. Each section is divided into five subsections (see the flow-chart in Figure 1.1). In subsections 2.1 and 3.1 we analyze the influence of inexact solution of inner systems or least squares on the maximum attainable accuracy in the outer iteration process for solving (1.2) or (1.3), and we estimate the ultimate norms of the true residuals $-B^T A^{-1} f + B^T A^{-1} B \bar{y}_k$ and $(I - \Pi) f - (I - \Pi) A (I - \Pi) \bar{x}_k$. In the consequent three subsections of sections 2 and 3, we give bounds for the ultimate norm of the true residuals $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T \bar{x}_k$. As we will see in subsections 2.2–2.4 and 3.2–3.4, the limiting accuracy of these residuals may significantly differ for various back-substitution formulas for computing $x_k$ or $y_k$, respectively. Subsections 2.5 and 3.5 contain forward analysis with the bounds for the errors $x - \bar{x}_k$ and $y - \bar{y}_k$. Throughout this paper our theoretical results are illustrated on the model example taken from [30]: we put $n = 100$, $m = 20$, and

$$A = \mathrm{tridiag}(1, 4, 1) \in \mathbb{R}^{n \times n}, \quad B = \mathrm{rand}(n, m), \quad f = \mathrm{rand}(n, 1).$$

The spectrum of $A$ and singular values of $B$ lie in the interval $[2.0010, 5.9990]$ and $[2.1727, 7.1695]$, respectively. Therefore the conditioning of $A$ or $B$ does not play an important role in our experiments. For further discussion, we refer to subsections 2.5 and 3.5.

For distinction, we denote quantities computed in finite precision arithmetic by bars. We assume that the usual rules of a well-designed floating-point arithmetic hold and use occasionally the notation $\mathrm{fl}(\cdot)$ for a computed result of an expression. The roundoff unit is denoted by $u$. In particular, for a matrix-vector multiplication the bound $\|\mathrm{fl}(Ax)-Ax\| \leq O(u)\|A\|\|x\|$ is used and $\|x\|$ denotes the 2-norm of the vector $x$; for a general matrix $A$ we make use of the spectral norm $\|A\|$ and the corresponding condition number $\kappa(A) = \|A\|/\sigma_{min}(A)$, where $\sigma_{min}(A)$ is the minimal singular value of $A$. For a symmetric positive definite matrix $A$, $\|x\|_A$ denotes the $A$-norm of the vector $x$. Finally, we apply the $O$-notation when suitable.

**2. Schur complement reduction method.** In this section we will discuss algorithms which compute simultaneously approximations $x_k$ and $y_k$ to the unknowns $x$ and $y$ and ideally fulfill the first block equation in (1.1)

$$Ax_k + By_k = f. \tag{2.1}$$

Our goal here is not to survey all existing schemes based on (2.1) but to analyze the numerical behavior of three implementations which use different back-substitution formulas for computing the approximate solution $x_k$. More precisely, without specifying any particular method, we assume that we have computed the approximate solution $y_{k+1}$ and the residual vector $r_{k+1}^{(y)}$ using the recursions

$$y_{k+1} = y_k + \alpha_k p_k^{(y)}, \tag{2.2}$$

$$r_{k+1}^{(y)} = r_k^{(y)} + \alpha_k B^T A^{-1} B p_k^{(y)} \tag{2.3}$$

with $r_0^{(y)} = -B^T A^{-1}(f - By_0)$. We will distinguish between the following three mathematically equivalent formulas:

$$x_{k+1} = x_k + \alpha_k(-A^{-1}Bp_k^{(y)}), \tag{2.4}$$

$$x_{k+1} = A^{-1}(f - By_{k+1}), \tag{2.5}$$

$$x_{k+1} = x_k + A^{-1}(f - Ax_k - By_{k+1}). \tag{2.6}$$

The resulting schemes are summarized in Figure 2.1. These schemes have been used and studied in the context of many applications, including various classical Uzawa algorithms, the two-level pressure correction approach, and the inner-outer iteration method for solving (1.1); see, e.g., the schemes with (2.4) in [29, 3], (2.5) in [15], or (2.6) in [11, 12, 4, 37], respectively. Because the solves with matrix $A$ in formulas (2.4)–(2.6) are expensive, these systems are in practice solved only approximately. Our analysis is based on the assumption that every solution of a symmetric positive definite system with the matrix $A$ is replaced by an approximate solution produced by an arbitrary method. The resulting vector is then interpreted as an exact solution of the system with the same right-hand side vector but with a perturbed matrix $A + \Delta A$. We always require that the relative norm of the perturbation is bounded as $\|\Delta A\| \leq \tau\|A\|$, where $\tau$ represents a backward error associated with the computed solution vector. We will always assume that the perturbation $\Delta A$ does not exceed the limitation given by the distance of $A$ to the nearest singular matrix and put restriction in the form $\tau\kappa(A) \ll 1$. It follows then from the standard perturbation analysis (see, e.g., [23, 6]) that

$$\|(A + \Delta A)^{-1} - A^{-1}\| \leq \frac{\tau\kappa(A)}{1 - \tau\kappa(A)}\|A^{-1}\|.$$

```
outer iteration

$y_0$, solve $Ax_0 = f - By_0$, $r_0^{(y)} = -B^T x_0$

for $k = 0, 1, 2, \dots$
    $y_{k+1} = y_k + \alpha_k p_k^{(y)}$

    inner iteration / back-substitution

        solve $Ap_k^{(x)} = -Bp_k^{(y)}$
    (A) $x_{k+1} = x_k + \alpha_k p_k^{(x)}$
    (B) solve $Ax_{k+1} = f - By_{k+1}$
    (C) solve $Au_k = f - Ax_k - By_{k+1}$, $x_{k+1} = x_k + u_k$

    $r_{k+1}^{(y)} = r_k^{(y)} - \alpha_k B^T p_k^{(x)}$
```

FIG. 2.1. *Schur complement reduction: Three different schemes for computing the approximate solution* $x_{k+1}$ *(called in the text the updated approximate solution* (A), *the approximate solution computed by a direct substitution* (B), *and the approximate solution computed by a corrected direct substitution* (C), *respectively).*

Note that if $\tau = O(u)$, then we have a backward stable method for solving the positive definite system with $A$. In our numerical experiments, we solve the systems with $A$ inexactly using the conjugate gradient method or with the Cholesky factorization as indicated by the notation $\tau = O(u)$.

**2.1. The attainable accuracy in the Schur complement system.** In this subsection we look at the ultimate accuracy in the outer iteration process by means of the true residual $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$. It is clear that if we perturb the Schur complement system $-B^TA^{-1}By = -B^TA^{-1}f$ to $-B^T(A+\Delta A)^{-1}B\hat{y} = -B^TA^{-1}f$, where $\|\Delta A\| \leq \tau\|A\|$, then the residual associated with $\hat{y}$ can be bounded as

$$\| - B^TA^{-1}f + B^TA^{-1}B\hat{y}\| \leq \frac{\tau\kappa(A)}{1-\tau\kappa(A)}\|A^{-1}\|\|B\|^2\|\hat{y}\|. \tag{2.7}$$

We see from (2.7) that there is a limitation to the accuracy of the residual obtained directly from $\hat{y}$ and its bound is proportional to $\tau$. Note that these considerations were made assuming exact arithmetic. The effects of rounding errors on the same quantity have been studied by Greenbaum [22], who considered a general class of methods for solving the fixed system of linear equations using two-term recursions given by (2.2) and (2.3). Using a similar approach we can extend these results and formulate the following theorem.

THEOREM 2.1. *The gap between the true residual* $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$ *and the updated residual* $\bar{r}_k^{(y)}$ *can be bounded as*

$$\|-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k - \bar{r}_k^{(y)}\| \leq \frac{[(2k+1)\tau + O(u)]\kappa(A)}{1-\tau\kappa(A)}\|A^{-1}\|\|B\|(\|f\|+\|B\|\bar{Y}_k),$$

*where* $\bar{Y}_k$ *is defined as a maximum norm over all computed approximate solutions* $\bar{Y}_k \equiv \max_{i=0,\dots,k} \|\bar{y}_i\|$.

*Proof.* The initial residual $\bar{r}_0^{(y)}$ is computed as $\bar{r}_0^{(y)} = -\mathrm{fl}(B^T\bar{x}_0)$, where $(A + \Delta A_0)\bar{x}_0 = \mathrm{fl}(f - By_0)$, $\|\Delta A_0\| \le \tau\|A\|$. It is easy to see that the statement holds for $k = 0$. The computed approximate solution $\bar{y}_{k+1}$ and the residual $\bar{r}_{k+1}^{(y)}$ satisfy

$$\bar{y}_{k+1} = \bar{y}_k + \bar{\alpha}_k\bar{p}_k^{(y)} + \Delta y_{k+1},\ \|\Delta y_{k+1}\| \le u\|\bar{y}_k\| + (2u+u^2)\|\bar{\alpha}_k\bar{p}_k^{(y)}\|, \tag{2.8}$$

$$\bar{r}_{k+1}^{(y)} = \bar{r}_k^{(y)} - \bar{\alpha}_k B^T\bar{p}_k^{(x)} + \Delta r_{k+1}^{(y)},\ \|\Delta r_{k+1}^{(y)}\| \le u\|\bar{r}_k^{(y)}\| + O(u)\|B\|\|\bar{\alpha}_k\bar{p}_k^{(x)}\|, \tag{2.9}$$

where $\bar{p}_k^{(x)}$ is the exact solution of the perturbed system

$$(A + \Delta A_k)\bar{p}_k^{(x)} = -\mathrm{fl}(B\bar{p}_k^{(y)}),\ \|\Delta A_k\| \le \tau\|A\|. \tag{2.10}$$

Multiplying (2.8) by $B^TA^{-1}B$, substituting (2.10) into the recurrence (2.9), and subtracting these two equations we get the recurrence

$$\begin{aligned} -B^TA^{-1}f + B^TA^{-1}B\bar{y}_{k+1} - \bar{r}_{k+1}^{(y)} &= -B^TA^{-1}f + B^TA^{-1}B\bar{y}_k - \bar{r}_k^{(y)} \\ &\quad -\bar{\alpha}_k(B^T\bar{p}_k^{(x)} + B^TA^{-1}B\bar{p}_k^{(y)}) + B^TA^{-1}B\Delta y_k - \Delta r_k^{(y)}. \end{aligned}$$

The norm of the vector $\bar{\alpha}_k\bar{p}_k^{(y)}$ can be bounded as $\|\bar{\alpha}_k\bar{p}_k^{(y)}\| \le \|\bar{y}_{k+1}\| + \|\bar{y}_k\| + \|\Delta y_{k+1}\|$. This bound in combination with (2.8) gives $\|\Delta y_{k+1}\| \le O(u)\bar{Y}_{k+1}$ and $\|\bar{\alpha}_k\bar{p}_k^{(y)}\| \le 3\bar{Y}_{k+1}$ which also implies

$$\|\bar{\alpha}_k\bar{p}_k^{(x)}\| \le \frac{3\|A^{-1}\|}{1 - \tau\kappa(A)}\|B\|\bar{Y}_{k+1}. \tag{2.11}$$

Using (2.10), the bound on $\|\bar{\alpha}_k\bar{p}_k^{(y)}\|$, and some elementary manipulation, we can estimate the term $\bar{\alpha}_k(B^T\bar{p}_k^{(x)} + B^TA^{-1}B\bar{p}_k^{(y)})$

$$\begin{aligned} \|\bar{\alpha}_k(B^T\bar{p}_k^{(x)} + B^TA^{-1}B\bar{p}_k^{(y)})\| &\le \|\bar{\alpha}_k B^T[(A+\Delta A_k)^{-1} - A^{-1}]\mathrm{fl}(B\bar{p}_k^{(y)})\| \\ +\|\bar{\alpha}_k B^TA^{-1}[\mathrm{fl}(B\bar{p}_k^{(y)}) - B\bar{p}_k^{(y)}]\| &\le \frac{[\tau + O(u)]\kappa(A)}{1 - \tau\kappa(A)}\|A^{-1}\|\|B\|^2\bar{Y}_{k+1}. \end{aligned}$$

Considering (2.9), (2.11), and the induction assumption on $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k - \bar{r}_k^{(y)}$ (similar to the one used in [22]), we obtain the bound for the error vector $\Delta r_{k+1}^{(y)}$ in the form

$$\|\Delta r_{k+1}^{(y)}\| \le \frac{O(u)\kappa(A)}{1 - \tau\kappa(A)}\|A^{-1}\|\|B\|(\|f\| + \|B\|\bar{Y}_{k+1})$$

which proves the statement of the theorem. □

It is a well-known fact that the residual $\bar{r}_k^{(y)}$ computed recursively via (2.3) usually converges far below $O(u)$. Using this assumption we can obtain from the estimate for the gap $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k - \bar{r}_k^{(y)}$ the estimate for the maximum attainable accuracy of the true residual $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$ itself. Summarizing, while the updated residual $\bar{r}_k^{(y)}$ converges to zero the true residual stagnates at the level proportional to $\tau$. This is also illustrated in our numerical example, where the Schur complement system $-B^TA^{-1}By = -B^TA^{-1}f$ is solved using the steepest descent method with the initial approximation $y_0$ set to zero. In Figure 2.2(a) we show the relative norms of the true residual $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$ (solid lines) and the updated residual $\bar{r}_k^{(y)}$ (dashed lines).

Similar to Greenbaum [22], we have shown that the gap between the true and updated residual is proportional to the maximum norm of approximate solutions computed during the whole iteration process. Since the Schur complement system is symmetric negative definite, the norm of the error or residual converges monotonically for the most iterative methods like the steepest descent, the conjugate gradient, the conjugate residual method, or other error/residual minimizing methods or at least becomes orders of magnitude smaller than the initial error/residual without exceeding this limit. In such cases, the quantity $\bar{Y}_k$ does not play an important role in the bound, and it can usually be replaced by $\|y_0\|$ or a small multiple of $\|y\|$. The situation is more complicated when $A$ is nonsingular and nonsymmetric; see [24].

As we already noted, the main difference with respect to the analysis of Greenbaum is that the floating-point multiplication with the fixed $A^{-1}$ is replaced by the step-dependent inexact solution of the system with $A$ such that it can be interpreted as the exact application of the matrix $(A+\Delta A_k)^{-1}$, where the perturbation matrix $\Delta A_k$ changes at every step $k$. This concept is very similar to the notion of inexact Krylov subspace methods (see [31] or [33]), which, on the other hand, do not take into account the effects of rounding errors. The theory of Greenbaum [22] could be directly applied if we only have at each iteration $\|\mathrm{fl}(B^TA^{-1}Bx) - B^TA^{-1}Bx\| \le O(u)\|A^{-1}\|\|B\|^2\|x\|$. Since in our idealized case $\mathrm{fl}(B^TA^{-1}Bx) = B^T(A+\Delta A_k)^{-1}Bx$ with $\|\Delta A_k\| \le \tau\|A\|$, we have only

$$\|\mathrm{fl}(B^TA^{-1}Bx) - B^TA^{-1}Bx\| \le \frac{\tau\kappa(A)}{1-\tau\kappa(A)}\|A^{-1}\|\|B\|^2\|x\|.$$

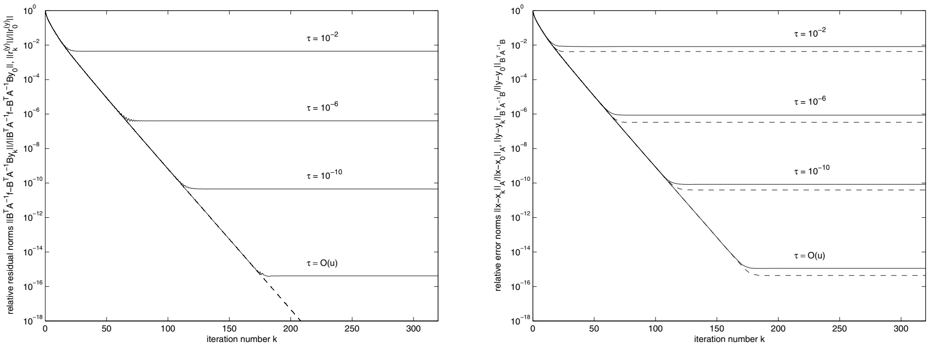
This bound could be improved if we make a restriction and use a variable tolerance for inner systems. If we require that every inner system is solved so that the relative residual of its computed solution needs the tolerance $\tau$, then every inexact application of the matrix $B^TA^{-1}B$ would satisfy the inequality

$$\|\mathrm{fl}(B^TA^{-1}Bx) - B^TA^{-1}Bx\| \le \tau\|A^{-1}\|\|B\|^2\|x\|. \tag{2.12}$$

Then the whole outer process (2.2) and (2.3) together with (2.12) could be interpreted as a floating-point iteration with the roundoff unit equal to $\tau$. The computation in this "extended" arithmetic would lead to

$$\| - B^TA^{-1}f + B^TA^{-1}B\bar{y}_k - \bar{r}_k^{(y)}\| \le \frac{O(\tau)}{1-\tau\kappa(A)}\|A^{-1}\|\|B\|^2(\|y\| + \bar{Y}_k).$$

A thorough rounding analysis of the block LU factorization has been given in [14] and further developed in the saddle point context in [26]. The approach was quite converse to the one used in our paper. It is assumed that all inner systems are solved in a backward stable way and the accuracy of computed approximate solutions is estimated in terms of the user prescribed tolerance for the outer Schur complement system. Roughly speaking, the higher tolerance $\eta$ leads to the higher level of attainable accuracy of the true residuals $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T\bar{x}_k$. This level is magnified by the quantities that play a similar role as the growth factor in the Gaussian elimination with partial pivoting (see, e.g., [23]). On the other hand, the parameter $\eta$ giving the threshold for the backward error cannot be infinitely small. Theorem 2.1 actually



(a) (b)

FIG. 2.2. *Schur complement reduction method:* (a) *the relative norms of the true residual* $-B^TA^{-1}f + B^TA^{-1}\bar{y}_k$ *(solid lines) and the updated residual* $\bar{r}_k^{(y)}$ *(dashed lines)—the updated solution scheme* (2.4); (b) *the relative error norms* $\|x - \bar{x}_k\|_A/\|x - \bar{x}_0\|_A$ *(solid lines) and* $\|y - \bar{y}_k\|_{B^TA^{-1}B}/\|y - y_0\|_{B^TA^{-1}B}$ *(dashed lines)—the updated solution scheme* (2.4).

gives its lower bound. Dividing the right-hand side by $\|A^{-1}\|\|B\|^2\|\bar{y}\|$ we end up with $\eta \geq O(u)\kappa(A)/(1 - O(u)\kappa(A))$.

In the following we will estimate the residuals $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T\bar{x}_k$. We will show that these quantities depend on the actual implementation of the back-substitution formula for $x_k$ and distinguish between three schemes (2.4), (2.5), and (2.6). No matter how we compute the approximations $\bar{x}_k$ and $\bar{y}_k$ it holds that

$$-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k = -B^T\bar{x}_k - B^TA^{-1}(f - A\bar{x}_k - B\bar{y}_k), \tag{2.13}$$

which gives the mutual relation between the residual $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$ in the Schur complement system and the residuals $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T\bar{x}_k$ associated with the saddle point system (1.1). According to Theorem 2.1, $\|-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k\|$ is ultimately $O(\tau)$. Then it is clear from (2.13) that both $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T\bar{x}_k$ cannot be proportional to the roundoff unit $u$. We will show that, depending on the chosen back-substitution scheme, we can ensure either that $f - A\bar{x}_k - B\bar{y}_k = O(\tau)$ with $-B^T\bar{x}_k = O(u)$ (scheme A (2.4)), or that $f - A\bar{x}_k - B\bar{y}_k = O(u)$ with $-B^T\bar{x}_k = O(\tau)$ (scheme C (2.6)), while the most straightforward scheme B (2.5) leads to both $f - A\bar{x}_k - B\bar{y}_k = O(\tau)$ and $-B^T\bar{x}_k = O(\tau)$.

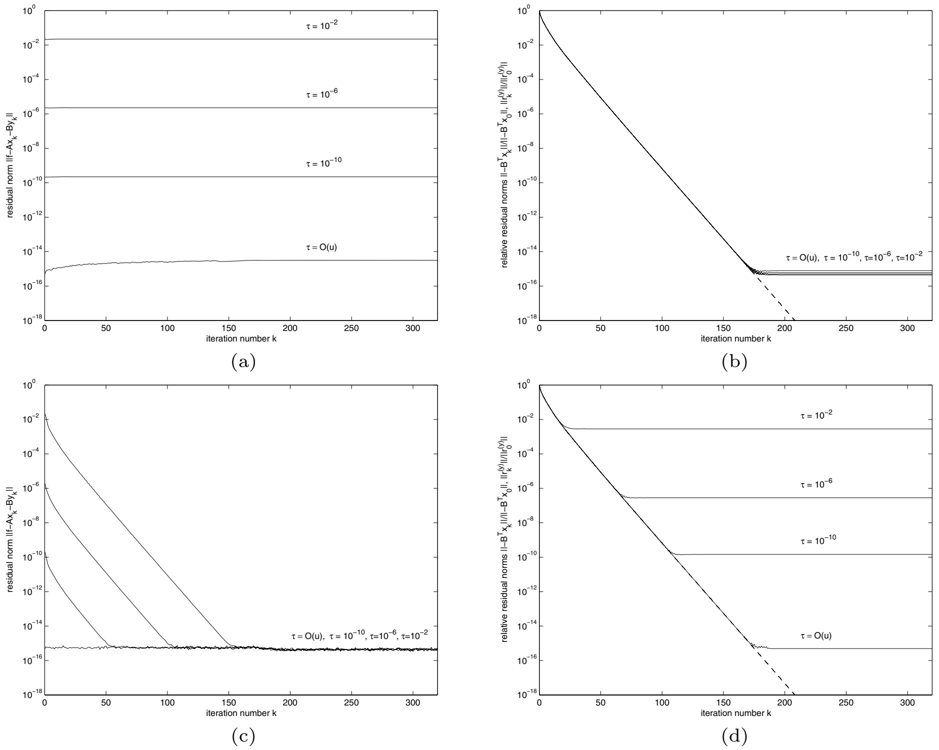
**2.2. Scheme A: The updated approximate solution.** In this subsection we analyze the generic update (2.4). It is clear that this scheme requires only one system solve with $A$ per iteration. Indeed, we compute only the direction vector $p_k^{(x)} = -A^{-1}Bp_k^{(y)}$, which appears in the recurrence $r_{k+1}^{(y)} = r_k^{(y)} - \alpha_k B^T p_k^{(x)}$ anyway. As we will see, in finite precision arithmetic this algorithm guarantees that $-B^T\bar{x}_k$ will ultimately reach $O(u)$. This happens despite the fact that the systems with the matrix block $A$ are computed inexactly with the parameter $\tau$ frequently much larger than $O(u)$.

THEOREM 2.2. *The true residual* $f - A\bar{x}_k - B\bar{y}_k$ *satisfies the bound*

$$\|f - A\bar{x}_k - B\bar{y}_k\| \leq O(u)(\|f\| + \|B\|\bar{Y}_k) + [(k+1)\tau + O(u)]\|A\|\bar{X}_k. \tag{2.14}$$

*The gap between the residuals* $-B^T\bar{x}_k$ *and* $\bar{r}_k^{(y)}$ *can be estimated as*

$$\|-B^T\bar{x}_k - \bar{r}_k^{(y)}\| \leq O(u)\|A^{-1}\|\|B\|(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k),$$



(a) (b)

(c) (d)

FIG. 2.3. *Schur complement reduction method:* (a) *the norms of the true residual* $f - A\bar{x}_k - B\bar{y}_k$ *and* (b) *the relative norms of the true residual* $-B^T\bar{x}_k$ *(solid lines) and the recursively computed residual* $\bar{r}_k^{(y)}$ *(dashed lines)—the updated solution scheme* (2.4); (c) *the norms of the true residual* $f - A\bar{x}_k - B\bar{y}_k$*—the corrected direct substitution scheme* (2.6); (d) *the relative norms of the true residual* $-B^T\bar{x}_k$ *(solid lines) and the recursively computed residual* $\bar{r}_k^{(y)}$ *(dashed lines)—the direct substitution scheme* (2.5).

*where* $\bar{X}_k$ *is now defined as a maximum norm over all computed approximate solutions* $\bar{X}_k \equiv \max_{i=0,\dots,k} \|\bar{x}_i\|$.

*Proof.* The computed approximate solution $\bar{x}_{k+1}$ satisfies

$$\bar{x}_{k+1} = \bar{x}_k + \bar{\alpha}_k \bar{p}_k^{(x)} + \Delta x_{k+1}, \ \|\Delta x_{k+1}\| \le u\|\bar{x}_k\| + (2u+u^2)\|\bar{\alpha}_k \bar{p}_k^{(x)}\|. \tag{2.15}$$

Substituting recurrently (2.15) and (2.8) into the residual

$$f - A\bar{x}_{k+1} - B\bar{y}_{k+1} = f - A\bar{x}_k - B\bar{y}_k - \bar{\alpha}_k(A\bar{p}_k^{(x)} + B\bar{p}_k^{(y)}) - A\Delta x_{k+1} - B\Delta y_{k+1},$$

we obtain the following bound:

$$\begin{aligned}\|f - A\bar{x}_k - B\bar{y}_k\| &\le \|f - A\bar{x}_0 - By_0\| \\ &+ \sum_{i=0}^{k-1} \left( \|\bar{\alpha}_i(A\bar{p}_i^{(x)} + B\bar{p}_i^{(y)})\| + \|A\|\|\Delta x_{i+1}\| + \|B\|\|\Delta y_{i+1}\| \right).\end{aligned}$$

Here we, in fact, reformulate the main result of Greenbaum [22, Theorem 2.2] and heavily use the fact that the vectors $\bar{p}_k^{(x)}$ satisfy the perturbed system (2.10). From Theorem 2.1 we have bounds $\|\Delta y_{k+1}\| \le O(u)\bar{Y}_{k+1}$ and $\|\bar{\alpha}_k \bar{p}_k^{(y)}\| \le 3\bar{Y}_{k+1}$ which also imply the bound (2.11). Using all of these results we get

$$\|\bar{\alpha}_k(A\bar{p}_k^{(x)} + B\bar{p}_k^{(y)})\| \le \|\bar{\alpha}_k[\mathrm{fl}(B\bar{p}_k^{(y)}) - B\bar{p}_k^{(y)}]\| + \|\Delta A_k\|\|\bar{\alpha}_k\bar{p}_k^{(x)}\|.$$

Further we use $\|\Delta x_{k+1}\| \le O(u)\bar{X}_{k+1}$ and $\|\bar{\alpha}_k\bar{p}_k^{(x)}\| \le 3\bar{X}_{k+1}$. Summarizing, we get the first result. The gap between $-B^T\bar{x}_{k+1}$ and $\bar{r}_{k+1}^{(y)}$ is equal to

$$-B^T\bar{x}_{k+1} - \bar{r}_{k+1}^{(y)} = -B^T\bar{x}_k - \bar{r}_k^{(y)} - B^T\Delta x_{k+1} - \Delta r_{k+1}^{(y)}$$

and it leads to the expansion containing just the local errors $\Delta x_{i+1}$, $\Delta y_{i+1}$ and the initial gap $-B^T\bar{x}_0 - \bar{r}_0^{(y)}$

$$-B^T\bar{x}_k - \bar{r}_k^{(y)} = -B^T\bar{x}_0 - \bar{r}_0^{(y)} - \sum_{i=0}^{k-1} B^T\Delta x_{i+1} - \sum_{i=0}^{k-1} \Delta r_{k+1}^{(y)}.$$

Taking norms, considering the bounds on $\|\Delta x_{k+1}\|$, $\|\Delta y_{k+1}\|$, (2.9), and the relation $\bar{r}_0^{(y)} = -\mathrm{fl}(B^T\bar{x}_0)$, we get the second result. □

As we will see in the next subsection, the bound for the gap $-B^T\bar{x}_k - \bar{r}_k^{(y)}$ is considerably better than for the scheme (2.5). In contrast to (2.18), it does not depend on $\tau$. Provided that $\bar{r}_k^{(y)}$ converges to zero, the true residual $-B^T\bar{x}_k$ will stagnate at the level proportional to $u$ and the second block equation in (1.1) will be satisfied to working accuracy.

Figures 2.3(a), (b) show the norms of the true residual $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T\bar{x}_k$ (solid lines), respectively, including the norms of the updated residual $\bar{r}_k^{(y)}$ (dashed lines). The numerical results are in good agreement with Theorem 2.2. The residual $f - A\bar{x}_k - B\bar{y}_k$ is growing slightly due to the accumulation of errors in inner systems $Ap_k^{(x)} = -Bp_k^{(y)}$ but it essentially remains on the level proportional to $\tau$. The residual $-B^T\bar{x}_k$ ultimately stagnates at $O(u)$. The formula (2.4) is suitable whenever the second block equation in (1.1) must be satisfied accurately, no matter how small or big the inner tolerance $\tau$ is.

### 2.3. Scheme B: The approximate solution computed by a direct substitution.

In this subsection we assume that $x_k$ is computed by the direct substitution (2.5). The computed $\bar{x}_k$ then satisfies the equality

$$(A + \Delta A_k)\bar{x}_k = \mathrm{fl}(f - B\bar{y}_k), \ \|\Delta A_k\| \le \tau\|A\|. \tag{2.16}$$

The perturbation matrices $\Delta A_k$ are different from those defined in subsection 2.1, but for simplicity we will keep the same notation. In the following we will show that the residual $\bar{r}_k^{(y)}$ is a good approximation for the residual $-B^T\bar{x}_k$, provided that they are above the level given by the bound for $-B^T\bar{x}_k - \bar{r}_k^{(y)}$. This quantity is now, however, proportional to $\tau$.

Theorem 2.3. *The true residual $f - A\bar{x}_k - B\bar{y}_k$ satisfies the bound*

$$\|f - A\bar{x}_k - B\bar{y}_k\| \le O(u)(\|f\| + \|B\|\|\bar{y}_k\|) + \tau\|A\|\|\bar{x}_k\|. \tag{2.17}$$

*The gap between the residuals $-B^T\bar{x}_k$ and $\bar{r}_k^{(y)}$ can be bounded as follows:*

$$\begin{aligned} \| - B^T\bar{x}_k - \bar{r}_k^{(y)}\| &\le O(u)\|A^{-1}\|\|B\|(\|f\| + \|B\|\bar{Y}_k) \\ &\quad + [(k+3)\tau + O(u)]\kappa(A)\|B\|\bar{X}_k, \end{aligned} \tag{2.18}$$

*where* $\bar{X}_k$ *is defined as* $\bar{X}_k \equiv \max_{i=0,\dots,k-1}\{\|\bar{x}_0\|, \|\bar{x}_k\|, \|\bar{\alpha}_i\bar{p}_i^{(x)}\|\}$.

*Proof.* The first result follows from (2.16) and the relation for the true residual

$$f - A\bar{x}_k - B\bar{y}_k = f - B\bar{y}_k - \mathrm{fl}(f - B\bar{y}_k) - \Delta A_k\bar{x}_k.$$

For the gap between $-B^T\bar{x}_k$ and $\bar{r}_k^{(y)}$ we have the identity

$$\begin{aligned}-B^T\bar{x}_k - \bar{r}_k^{(y)} = {}& -B^TA^{-1}f + B^TA^{-1}B\bar{y}_k - \bar{r}_k^{(y)} + B^TA^{-1}\Delta A_k\bar{x}_k\\ &+ B^TA^{-1}[\mathrm{fl}(f - B\bar{y}_k) - (f - B\bar{y}_k)].\end{aligned} \tag{2.19}$$

The statement of Theorem 2.1 together with (2.19) gives the second result (2.18). ☐

Indeed, while the residual $\bar{r}_k^{(y)}$ converges ultimately below $O(u)$, the residual $-B^T\bar{x}_k$ will remain proportional to $\tau$. The norm of $f - A\bar{x}_k - B\bar{y}_k$ is unconditionally bounded by the term proportional to $\tau$ dominating other terms in (2.17).

Figure 2.3(d) shows the norms of $-B^T\bar{x}_k$ (solid lines) and $\bar{r}_k^{(y)}$ (dashed lines). The residual $f - A\bar{x}_k - B\bar{y}_k$ behaves similarly to that of the scheme (2.4) shown in plot (a). The residual $f - A\bar{x}_k - B\bar{y}_k$ remains almost constant since it is nothing but the residual of the system $Ax_k = f - By_k$ solved in each iteration with the uniform accuracy.

### 2.4. Scheme C: The approximate solution computed with a corrected direct substitution.

The third back-substitution formula (2.6) can be derived by a correction of the scheme (2.5) and requires two system solves with $A$. In this subsection we show that its numerical behavior is very similar to the behavior of classical nonstationary iterative methods described and analyzed by Higham [23]. We prove that under certain conditions the true residual $f - A\bar{x}_k - B\bar{y}_k$ ultimately converges to the level proportional to $u$, which is significantly smaller than those residuals for the previous two schemes.

THEOREM 2.4. *Assume for sufficiently large* $k$ *with* $\|\bar{y}_{k+1} - \bar{y}_k\| \le O(u)\bar{Y}_{k+1}$ *that there exists a step* $k_0$ *such that the true residual* $f - A\bar{x}_k - B\bar{y}_k$ *is bounded by*

$$\|f - A\bar{x}_k - B\bar{y}_k\| \le O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k) \tag{2.20}$$

*for all steps* $k \ge k_0$. *The gap between* $-B^T\bar{x}_k$ *and* $\bar{r}_k^{(y)}$ *can be estimated as follows*:

$$\|-B^T\bar{x}_k - \bar{r}_k^{(y)}\| \le O(u)\|A^{-1}\|\|B\|(\|f\| + \|B\|\bar{Y}_k) + [(k+3)\tau + O(u)]\kappa(A)\|B\|\bar{X}_k.$$

*The quantity* $\bar{X}_k$ *is here defined as* $\bar{X}_k \equiv \max_{i=0,\dots,k-1}\{\|\bar{x}_0\|, \|\bar{x}_k\|, \|\bar{\alpha}_i\bar{p}_i^{(x)}\|\}$.

*Proof.* The computed approximate solution $\bar{x}_{k+1}$ satisfies

$$\bar{x}_{k+1} = \bar{x}_k + \bar{u}_k + \Delta x_{k+1}, \quad \|\Delta x_{k+1}\| \le u(\|\bar{x}_k\| + \|\bar{u}_k\|), \tag{2.21}$$

where the vector $\bar{u}_k$ is the exact solution of the system

$$(A + \Delta A_{k+1})\bar{u}_k = \mathrm{fl}(f - A\bar{x}_k - B\bar{y}_{k+1}), \quad \|\Delta A_{k+1}\| \le \tau\|A\|. \tag{2.22}$$

The residual $f - A\bar{x}_{k+1} - B\bar{y}_{k+1}$ can be expressed using (2.21) and (2.22) as

$$\begin{aligned}f - A\bar{x}_{k+1} - B\bar{y}_{k+1} &= \Delta A_{k+1}\bar{u}_k - A\Delta x_{k+1}\\ &\quad + \mathrm{fl}(f - A\bar{x}_k - B\bar{y}_{k+1}) - (f - A\bar{x}_k - B\bar{y}_{k+1})\\ &= G_{k+1}(f - A\bar{x}_k - B\bar{y}_k) - G_{k+1}B(\bar{\alpha}_k\bar{p}_k^{(y)}) + h_{k+1},\end{aligned} \tag{2.23}$$

where the matrix $G_{k+1}$ and the vector $h_{k+1}$ are defined as $G_{k+1} \equiv \Delta A_{k+1}(A + \Delta A_{k+1})^{-1}$ and $h_{k+1} \equiv (I + G_{k+1})[\mathrm{fl}(f - A\bar{x}_k - B\bar{y}_{k+1}) - (f - A\bar{x}_k - B\bar{y}_{k+1})] - A\Delta x_{k+1} - G_{k+1}B\Delta y_{k+1}$. From a recursive use of the formula (2.23) we obtain

$$f - A\bar{x}_k - B\bar{y}_k = G_k \cdots G_1(f - A\bar{x}_0 - By_0) - \sum_{i=0}^{k-1} G_k \cdots G_{i+2}(G_{i+1}B\bar{\alpha}_i\bar{p}_i^{(y)} - h_{i+1}).$$

Taking norms, using the relation $\|\bar{\alpha}_i\bar{p}_i^{(y)}\| \le \|\bar{y}_{i+1} - \bar{y}_i\| + \|\Delta y_{i+1}\|$ and $\|\Delta A_i\| \le \tau\|A\|$ we obtain the uniform bound $\|G_i\| \le \tau\kappa(A)[1 - \tau\kappa(A)]^{-1} < 1$. This leads to the inequality

$$\begin{aligned}\text{(2.24)}\qquad \|f - A\bar{x}_k - B\bar{y}_k\| \le{}& \left(\frac{\tau\kappa(A)}{1 - \tau\kappa(A)}\right)^k \|f - A\bar{x}_0 - By_0\| \\ &+ \sum_{i=0}^{k-1}\left(\frac{\tau\kappa(A)}{1 - \tau\kappa(A)}\right)^{k-i} \|B\|\|\bar{y}_{i+1} - \bar{y}_i\| \\ &+ k \max_{i=0,\ldots,k-1} \|h_{i+1}\| + k \max_{i=0,\ldots,k-1} \|B\|\|\Delta y_{i+1}\|.\end{aligned}$$

For the vector $h_{k+1}$ it further follows that

$$\|h_{k+1}\| \le O(u)[\|f\| + \|A\|(\|\bar{x}_{k+1}\| + \|\bar{x}_k\|) + \|B\|\bar{Y}_{k+1}].$$

It is easy to see that for sufficiently large $k$ the first term on the right-hand side of (2.24) will decrease far below $O(u)$, while the second term will be at most $O(u)\|B\|\bar{Y}_{k+1}$ for all steps $k$ starting from some index $k_0$. Summarizing, for sufficiently large $k \ge k_0$ we have the bound

$$\|f - A\bar{x}_k - B\bar{y}_k\| \le O(u)[\|f\| + \|A\|(\|\bar{x}_{k+1}\| + \|\bar{x}_k\|) + \|B\|\bar{Y}_k].$$

The second statement can be proved considering

$$\begin{aligned}-B^T\bar{x}_{k+1} - \bar{r}_{k+1}^{(y)} ={}& -B^TA^{-1}f + B^TA^{-1}B\bar{y}_{k+1} - \bar{r}_{k+1}^{(y)} \\ &- B^T[(A + \Delta A_{k+1})^{-1} - A^{-1}]\mathrm{fl}(f - A\bar{x}_k - B\bar{y}_{k+1}) \\ &- B^TA^{-1}[\mathrm{fl}(f - A\bar{x}_k - B\bar{y}_{k+1}) - (f - A\bar{x}_k - B\bar{y}_{k+1})].\end{aligned}$$

The first term on the right-hand side can be estimated using Theorem 2.1. Based on (2.22) we have

$$\|[(A + \Delta A_{k+1})^{-1} - A^{-1}]\mathrm{fl}(f - A\bar{x}_k - B\bar{y}_{k+1})\| \le \frac{\tau\kappa(A)}{1 - \tau\kappa(A)}\|\bar{u}_k\|,$$

which together with the bound on $\|\bar{u}_k\|$ completes the proof. □

In Theorem 2.4 we assume that $\bar{y}_k$ ultimately stagnates so that $\|\bar{y}_{k+1} - \bar{y}_k\| \le O(u)\bar{Y}_{k+1}$ for sufficiently large $k \ge k_0$. It appears that this condition does not represent a serious restriction. Using (2.8) we have $\|\bar{y}_{k+1} - \bar{y}_k\| \le \|\bar{\alpha}_k\bar{p}_k^{(y)}\| + O(u)\bar{Y}_{k+1}$. We will show that the norm of $\bar{\alpha}_k\bar{p}_k^{(y)}$ is much smaller than $u$ for large $k$, i.e., we can absorb it into the term $O(u)\bar{Y}_{k+1}$. Denoting $\hat{S}_k \equiv B^T(A + \Delta A_k)^{-1}B$, using (2.9) and (2.10) we have the bound

$$\|\bar{\alpha}_k\bar{p}_k^{(y)}\| \le 2\|\hat{S}_k^{-1}\|(\|\bar{r}_{k+1}^{(y)}\| + \|\bar{r}_k^{(y)}\|) + O(u)\|\hat{S}_k^{-1}\|\|(A + \Delta A_k)^{-1}\|\|B\|^2\|\bar{\alpha}_k\bar{p}_k^{(y)}\|.$$

Provided that $O(u)\|\hat{S}_k^{-1}\|\|(A+\Delta A_k)^{-1}\|\|B\|^2 < 1$, we obtain

$$\|\bar{\alpha}_k \bar{p}_k^{(y)}\| \leq \frac{2\|\hat{S}_k^{-1}\|(\|\bar{r}_{k+1}^{(y)}\| + \|\bar{r}_k^{(y)}\|)}{1 - O(u)\|\hat{S}_k^{-1}\|\|(A+\Delta A_k)^{-1}\|\|B\|^2}.$$

Since the norms of updated residuals decrease far below the roundoff unit, the assumption on $\|\bar{y}_{k+1} - \bar{y}_k\|$ will be true for sufficiently large $k$. Note that $O(u)\|\hat{S}_k^{-1}\|\|(A+\Delta A_k)^{-1}\|\|B\|^2 < 1$ is nothing but the restricted assumption of numerical nonsingularity of the Schur complement matrix $B^T A^{-1} B$.

The bound (2.20) is significantly better than its counterparts (2.14) and (2.17). Theorem 2.4 describes that the residual $f - A\bar{x}_k - B\bar{y}_k$ will ultimately reach the roundoff unit level provided that the matrix $G_k G_{k-1} \cdots G_1$ converges to zero for $k \to \infty$. As soon as iterates $\bar{y}_k$ start to stagnate at their limiting accuracy level, the rate of convergence of this nonstationary iteration process is bounded by the factor $\tau\kappa(A)[1-\tau\kappa(A)]^{-1}$. The behavior of $-B^T\bar{x}_k$ is similar to that of scheme (2.5). Indeed, when $\bar{r}_k^{(y)}$ converges ultimately below $O(u)$, the residual $-B^T\bar{x}_k$ remains proportional to $\tau$. Figure 2.3(c) shows the norms of the residual $f - A\bar{x}_k - B\bar{y}_k$. The plot for $-B^T\bar{x}_k$ (not reported) is similar to the plot (d) for the scheme (2.5). It is clear that in our well-conditioned case the stationary method converges very fast and the rate of decrease of $f - A\bar{x}_k - B\bar{y}_k$ is essentially comparable to the convergence rate of the outer iteration.

**2.5. Forward error analysis.** In this subsection we estimate the maximum attainable accuracy in terms of the errors $x - \bar{x}_k$ and $y - \bar{y}_k$. First we formulate the bounds in the 2-norm, then in the $A$-norm of the error $x - \bar{x}_k$, and then in the $B^T A^{-1} B$-norm of the error $y - \bar{y}_k$. The errors $x - \bar{x}_k$ and $y - \bar{y}_k$, and the residuals $f - A\bar{x}_k - B\bar{y}_k$ and $-B^T\bar{x}_k$, satisfy

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} x - \bar{x}_k \\ y - \bar{y}_k \end{pmatrix} = \begin{pmatrix} f - A\bar{x}_k - B\bar{y}_k \\ -B^T\bar{x}_k \end{pmatrix}. \tag{2.25}$$

We have the explicit expression for the inverse of the saddle point matrix

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix}^{-1} = \begin{pmatrix} (I-\Pi)A^{-1} & -\Pi B(B^T B)^{-1} \\ -(B^T B)^{-1} B^T \Pi^T & -(B^T A^{-1} B)^{-1} \end{pmatrix},$$

where $\Pi \equiv A^{-1}B(B^T A^{-1} B)^{-1} B^T$ represents the oblique projector onto a range of $R(B)$ along $N(B^T)$. Considering (2.25), the inequalities $\|(I-\Pi)A^{-1}\| \leq \|A^{-1}\|$, and $\|A^{-1}B(B^T A^{-1}B)^{-1}\| = \|\Pi B(B^T B)^{-1}\| \leq \|(B^T B)^{-1}\|^{1/2}$ we obtain the bounds

$$\|x - \bar{x}_k\| \leq \gamma_1 \|f - A\bar{x}_k - B\bar{y}_k\| + \gamma_2 \| - B^T\bar{x}_k\|, \tag{2.26}$$

$$\|y - \bar{y}_k\| \leq \gamma_2 \|f - A\bar{x}_k - B\bar{y}_k\| + \gamma_3 \| - B^T\bar{x}_k\|, \tag{2.27}$$

where $\gamma_1 \equiv \sigma_{min}^{-1}(A)$, $\gamma_2 \equiv \sigma_{min}^{-1}(B)$, and $\gamma_3 \equiv \sigma_{min}^{-1}(B^T A^{-1} B)$ are constants independent of the iteration step $k$. It is clear from (2.26), (2.27) and Theorems 2.2, 2.3, and 2.4 that $\|x - \bar{x}_k\|$ and $\|y - \bar{y}_k\|$ will be $O(\tau)$ for all back-substitution schemes. In contrast to our numerical example, the saddle point systems that arise in practice can be ill-conditioned. In such cases the constants $\gamma_1$, $\gamma_2$, and $\gamma_3$ may play an important role.

In exact arithmetic we have $\|x-x_k\|_A = \|y-y_k\|_{B^TA^{-1}B}$. Since in finite precision arithmetic the residual $f - A\bar{x}_k - B\bar{y}_k$ is no longer zero, instead of this identity we get

$$\big|\|x-\bar{x}_k\|_A - \|y-\bar{y}_k\|_{B^TA^{-1}B}\big| \le \gamma_1^{1/2}\|f - A\bar{x}_k - B\bar{y}_k\|. \tag{2.28}$$

We can also formulate the proposition, which gives bounds for the errors in terms of the residuals $f - A\bar{x}_k - B\bar{y}_k$ and $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$.

Theorem 2.5. *The $A$-norm of the error $x-\bar{x}_k$ and the $B^TA^{-1}B$-norm of the error $y-\bar{y}_k$ can be bounded as*

$$\|x-\bar{x}_k\|_A \le \gamma_1^{1/2}\|f - A\bar{x}_k - B\bar{y}_k\| + \gamma_3^{1/2}\|-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k\|, \tag{2.29}$$

$$\|y-\bar{y}_k\|_{B^TA^{-1}B} \le \gamma_3^{1/2}\|-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k\|. \tag{2.30}$$

*Proof.* It follows from (2.28) that

$$\begin{aligned} \|x-\bar{x}_k\|_A &\le \|y-\bar{y}_k\|_{B^TA^{-1}B} + \big|\|x-\bar{x}_k\|_A - \|y-\bar{y}_k\|_{B^TA^{-1}B}\big| \\ &\le \|y-\bar{y}_k\|_{B^TA^{-1}B} + \sigma_{min}^{-1/2}(A)\|f - A\bar{x}_k - B\bar{y}_k\|. \end{aligned} \tag{2.31}$$

For the $B^TA^{-1}B$-norm of the error $y-\bar{y}_k$ we have

$$\|y-\bar{y}_k\|_{B^TA^{-1}B} = \|B^TA^{-1}f - B^TA^{-1}B\bar{y}_k\|_{(B^TA^{-1}B)^{-1}}, \tag{2.32}$$

which completes the proof. □

The first term on the right-hand side of (2.29) should be zero in exact arithmetic and it describes how well the computed $\bar{x}_k$ and $\bar{y}_k$ satisfy (2.1). The second term is related to the Schur complement residual which in exact arithmetic should converge to zero. The recursively computed residual $\bar{r}_k^{(y)}$ is a good approximation to $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$, provided they are above the level given by Theorem 2.1. Therefore its norm represents an easily computable quantity for the second term on the right-hand side of (2.29). The residual $f - A\bar{x}_k - B\bar{y}_k$ depends on the computed $\bar{x}_k$ and we distinguish between three schemes with (2.4), (2.5), and (2.6), respectively. We can see that, no matter which implementation we use, $-B^TA^{-1}f + B^TA^{-1}B\bar{y}_k$ is a dominating quantity in (2.29). Therefore, $\|x-\bar{x}_k\|_A$ can be thus well approximated during the convergence by the quantity $\gamma_3^{1/2}\|\bar{r}_k^{(y)}\|$ or its estimate. Similar can be said also for $\|y-\bar{y}_k\|_{B^TA^{-1}B}$; see (2.30).

The errors $x-\bar{x}_k$ and $y-\bar{y}_k$ can be estimated with more sophisticated but easily computable bounds (without explicit use of residuals and conditioning). As an example we refer to the rounding error analysis of the conjugate gradient method and various mathematically equivalent formulas for estimating $\|x-\bar{x}_k\|_A$ [32]. It appears that although many existing bounds were developed using exact arithmetic considerations, they estimate successfully the energy error using computed quantities which can be orders of magnitude different from their exact precision counterparts. Therefore, despite that we assume that $A^{-1}$ is performed inexactly, it is feasible to estimate the $B^TA^{-1}B$-norm of the error $y-\bar{y}_k$.

In Figure 2.2(b) we report the relative error norms $\|x-\bar{x}_k\|_A/\|x-\bar{x}_0\|_A$ and $\|y-\bar{y}_k\|_{B^TA^{-1}B}/\|y-y_0\|_{B^TA^{-1}B}$. The inverse of $A$ in the computation of the $B^TA^{-1}B$-norm is computed by a direct solver. In agreement with (2.29) and (2.30) and Theorems 2.2, 2.3, and 2.4 (see also Figure 2.3), the relative $A$-norm of the error $x-\bar{x}_k$ and also the relative $B^TA^{-1}B$-norm of the error $y-\bar{y}_k$ begin to stagnate at the level proportional to $\tau$. Since the behavior of these quantities for all implementations is similar, we present only the results for the scheme (2.5). The slight difference is visible only in the gap between both error norms given by the estimate (2.28).

## 3. Null-space projection method.

In this section we deal with algorithms which compute approximations $x_k$ and $y_k$ such that $x_k$ satisfies $B^T x_k = 0$ and $y_k$ solves the least squares problem minimizing the residual $f - Ax_k - By_k$, i.e.,

$$\|f - Ax_k - By_k\| = \min_{v \in \mathbb{R}^m} \|f - Ax_k - Bv\|. \tag{3.1}$$

We will denote (3.1) by $By_k \approx f - Ax_k$ and assume that the approximate solution $x_{k+1}$ and the residual vector $r_{k+1}^{(x)}$ are computed using

$$x_{k+1} = x_k + \alpha_k p_k^{(x)}, \tag{3.2}$$

$$r_{k+1}^{(x)} = r_k^{(x)} - \alpha_k A p_k^{(x)} - B p_k^{(y)}, \tag{3.3}$$

where $r_0^{(x)} = B^\dagger (f - Ax_0)$. The vectors $x_0$ and $p_k^{(x)}$ belong to $N(B^T)$ and $p_k^{(y)}$ solves the problem $Bp_k^{(y)} \approx r_k^{(x)} - \alpha_k A p_k^{(x)}$ minimizing the residual

$$\|r_k^{(x)} - \alpha_k A p_k^{(x)} - B p_k^{(y)}\| = \min_{p \in \mathbb{R}^m} \|r_k^{(x)} - \alpha_k A p_k^{(x)} - Bp\|.$$

This residual update strategy was proposed in [20] (see also [10, 9]) and is used to reduce the roundoff errors in the projection onto $N(B^T)$. Note that the vectors $p_k^{(y)}$ can be, with no additional cost, used as direction vectors for computing the approximate solution $y_{k+1}$. Again we will distinguish between three back-substitution formulas (the resulting schemes are described in Figure 3.1)

$$y_{k+1} = y_k + p_k^{(y)}, \; p_k^{(y)} = B^\dagger (r_k^{(x)} - \alpha_k A p_k^{(x)}), \tag{3.4}$$

$$y_{k+1} = B^\dagger (f - Ax_{k+1}), \tag{3.5}$$

$$y_{k+1} = y_k + B^\dagger (f - Ax_{k+1} - By_k). \tag{3.6}$$

The pseudoinverse $B^\dagger$ in (3.4)–(3.6) is applied by solving the least squares with the matrix $B$. These problems are solved inexactly. In our considerations we will assume that the computed solution $\bar{v}$ of the least squares problem $Bv \approx c$ is an exact solution of a perturbed problem $(B + \Delta B)\bar{v} \approx c + \Delta c$ with $\|\Delta B\|/\|B\| \leq \tau$ and $\|\Delta c\|/\|c\| \leq \tau$. The parameter $\tau$ again represents the measure for the inexact solution of the least squares with $B$ and actually describes the backward error. This can be achieved in many different ways considering the inner iteration loop solving the associated system of normal equations, the augmented system formulation, or solving it directly. Similar inexact schemes have been considered for solving quadratic programming problems [1, 2], multigrid methods [9, 10], or constraint preconditioners [25, 30, 28]. We assume $\tau\kappa(B) \ll 1$ which guarantees $B + \Delta B$ to have a full column rank. This allows the use of the perturbation theory (see [34] or [23, Lemma 19.8]), in particular the inequalities

$$\|(B + \Delta B)^\dagger\| \leq \frac{\|B^\dagger\|}{1 - \tau\kappa(B)}, \quad \|BB^\dagger - B(B + \Delta B)^\dagger\| \leq \frac{2\tau\kappa(B)}{1 - \tau\kappa(B)}.$$

Note that if $\tau = O(u)$, then we have a backward stable method for solving the least squares problem with $B$. In our experiments we applied the conjugate gradient least squares (CGLS) method [6] with the stopping criterion based on the corresponding backward error. Notation $\tau = O(u)$ stands for the Householder QR factorization.

```
outer iteration
$x_0$, solve $By_0 \approx f - Ax_0$, $r_0^{(x)} = f - Ax_0 - By_0$
for $k = 0, 1, 2, \dots$
    $x_{k+1} = x_k + \alpha_k p_k^{(x)}$
    inner iteration / back-substitution
        solve $Bp_k^{(y)} \approx r_k^{(x)} - \alpha_k A p_k^{(x)}$
        (A) $y_{k+1} = y_k + p_k^{(y)}$
        (B) solve $By_{k+1} \approx f - Ax_{k+1}$
        (C) solve $Bq_k \approx f - Ax_{k+1} - By_k$, $y_{k+1} = y_k + q_k$
    $r_{k+1}^{(x)} = r_k^{(x)} - \alpha_k A p_k^{(x)} - B p_k^{(y)}$
```

FIG. 3.1. *Null-space projection method: Three different schemes for computing the approximate solution* $y_{k+1}$ *(called in the text the updated approximate solution* (A), *the approximate solution computed by a direct substitution* (B), *and the approximate solution computed by a corrected direct substitution* (C), *respectively).*

**3.1. The attainable accuracy in the projected system.** In this subsection we look at the accuracy in the outer iteration for solving the projected system $(I - \Pi)A(I - \Pi)x = (I - \Pi)f$. We can consider the perturbed system

$$(I - \hat{\Pi})A(I - \hat{\Pi})\hat{x} = (I - \hat{\Pi})f, \tag{3.7}$$

where $\hat{\Pi} = (B + \Delta B)(B + \Delta B)^\dagger$ such that $\|\Delta B\| \leq \tau \|B\|$. The residual associated with the solution of (3.7) can be written as

$$(I - \Pi)f - (I - \Pi)A(I - \Pi)\hat{x} = (\hat{\Pi} - \Pi)f + (I - \hat{\Pi})A(\Pi - \hat{\Pi})\hat{x} + (\Pi - \hat{\Pi})A(I - \Pi)\hat{x}$$

and due to $\|\hat{\Pi} - \Pi\| \leq \|\Delta B\| \min\{\|B^\dagger\|, \|(B + \Delta B)^\dagger\|\}$ [23, Lemma 19.8], we have

$$\|(I - \Pi)f - (I - \Pi)A(I - \Pi)\hat{x}\| \leq \frac{2\tau\kappa(B)}{1 - \tau\kappa(B)}(\|f\| + \|A\|\|\hat{x}\|).$$

Indeed, even if we assume exact arithmetic, the residual obtained directly from $\hat{x}$ is proportional to the parameter $\tau$. In addition, we ideally have $(B + \Delta B)^T\hat{x} = 0$ which implies $\|-B^T\hat{x}\| \leq \tau\|B\|\|\hat{x}\|$. Therefore we can expect that also the residual $-B^T\bar{x}_k$ associated with the computed approximate solution $\bar{x}_k$ will be proportional to $\tau$. Such analysis is dependent on the choice of a particular method with the recurrences (3.2) and (3.3), and therefore we do not give it here. In accordance with [22] it seems reasonable that the bound for $-B^T\bar{x}_k$ is proportional to the factor $\bar{X}_k$. Moreover, the error in the projection of an arbitrary vector is represented in the bounds by $\tau\kappa(B)/[1 - \tau\kappa(B)]$. Therefore $-B^T\bar{x}_k$ and $\Pi\bar{x}_k$ can be expected to have the form

$$\|-B^T\bar{x}_k\| \leq \frac{O(\tau)\|B\|}{1 - \tau\kappa(B)}\bar{X}_k, \quad \|\Pi\bar{x}_k\| \leq \frac{O(\tau)\kappa(B)}{1 - \tau\kappa(B)}\bar{X}_k. \tag{3.8}$$

Theorem 3.1 shows that the true residual $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ is ultimately proportional to $\tau$, while its projection onto $N(B^T)$ will finally reach the level $O(u)$ provided that the updated residual $\bar{r}_k^{(x)}$ converges far below that level.

THEOREM 3.1. *The gap between the true residual* $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ *and the projection of the updated residual* $(I-\Pi)\bar{r}_k^{(x)}$ *can be bounded by*

$$\|(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k-(I-\Pi)\bar{r}_k^{(x)}\| \le \frac{O(\tau)\kappa(B)}{1-\tau\kappa(B)}(\|f\|+\|A\|\bar{X}_k),$$

*where* $\bar{X}_k \equiv \max_{i=0,\dots,k}\|\bar{x}_i\|$.

*Proof.* The computed approximation $\bar{x}_{k+1}$ satisfies the relations

$$\bar{x}_{k+1}=\bar{x}_k+\bar{\alpha}_k\bar{p}_k^{(x)}+\Delta x_{k+1},\ \|\Delta x_{k+1}\|\le u\|\bar{x}_k\|+(2u+u^2)\|\bar{\alpha}_k\bar{p}_k^{(x)}\|. \tag{3.9}$$

The inequality $\|\bar{\alpha}_k\bar{p}_k^{(x)}\|\le\|\bar{x}_{k+1}\|+\|\bar{x}_k\|+\|\Delta x_{k+1}\|$ gives $\|\bar{\alpha}_k\bar{p}_k^{(x)}\|\le 3\bar{X}_{k+1}$ and $\|\Delta x_{k+1}\|\le O(u)\bar{X}_{k+1}$. The vectors $\bar{y}_0$ and $\bar{p}_k^{(y)}$ satisfy $(B+\Delta B_0)\bar{y}_0\approx \mathrm{fl}(f-Ax_0)+\Delta c_0$ with $\|\Delta B_0\|\le\tau\|B\|$, $\|\Delta c_0\|\le\tau\|\mathrm{fl}(f-Ax_0)\|$, and

$$(B+\Delta B_k)\bar{p}_k^{(y)}\approx \mathrm{fl}(\bar{r}_k^{(x)}-\bar{\alpha}_kA\bar{p}_k^{(x)})+\Delta c_k, \tag{3.10}$$

$$\|\Delta B_k\|\le\tau\|B\|,\ \|\Delta c_k\|\le\tau\|\mathrm{fl}(\bar{r}_k^{(x)}-\bar{\alpha}_kA\bar{p}_k^{(x)})\|. \tag{3.11}$$

For updated residuals we have $\bar{r}_0^{(x)}=\mathrm{fl}(f-Ax_0-B\bar{y}_0)$ and

$$\bar{r}_{k+1}^{(x)}=\bar{r}_k^{(x)}-\bar{\alpha}_kA\bar{p}_k^{(x)}-B\bar{p}_k^{(y)}+\Delta r_{k+1}^{(x)}, \tag{3.12}$$

$$\|\Delta r_{k+1}^{(x)}\|\le O(u)(\|\bar{r}_k^{(x)}\|+\|A\|\|\bar{\alpha}_k\bar{p}_k^{(x)}\|+\|B\|\|\bar{p}_k^{(y)}\|). \tag{3.13}$$

The recursive use of (3.9) and (3.12) leads to the expression for the gap between the projections of $f-A\bar{x}_k$ and $\bar{r}_k^{(x)}$

$$(I-\Pi)(f-A\bar{x}_k-\bar{r}_k^{(x)})=(I-\Pi)(f-A\bar{x}_0-\bar{r}_0^{(x)})-\sum_{i=0}^{k-1}(I-\Pi)(A\Delta x_{i+1}+\Delta r_{i+1}^{(x)}).$$

Taking norms and corresponding bounds we get the following after some manipulation:

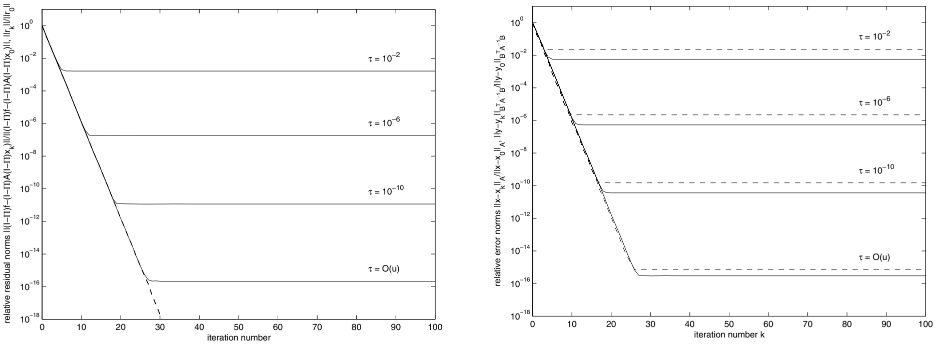
$$\|(I-\Pi)(f-A\bar{x}_k-\bar{r}_k^{(x)})\|\le\frac{O(u)\kappa(B)}{1-\tau\kappa(B)}\left(\|f\|+\|A\|\bar{X}_k\right). \tag{3.14}$$

Here we have used that $\|\bar{r}_k^{(x)}\|\le\|\bar{r}_0^{(x)}\|$ for $k=0,1,\dots$ which seems reasonable when solving the positive semidefinite problem. For the gap between $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ and $(I-\Pi)\bar{r}_k^{(x)}$, we can write

$$\|(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k-(I-\Pi)\bar{r}_k^{(x)}\|\le\|(I-\Pi)(f-A\bar{x}_k-\bar{r}_k^{(x)})\|+\|(I-\Pi)A\Pi\bar{x}_k\|.$$

Considering (3.14) and (3.8) we can conclude the proof. □

In Figure 3.2(a) we report the relative norms of the true residual $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ (solid lines) and the updated residual $\bar{r}_k^{(x)}$ (dashed lines). The numerical results confirm that the residual $f-A\bar{x}_k$ is within $N(B^T)$ approximated by $\bar{r}_k^{(x)}$ to the working precision $u$. However, this is not true for the residual $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ which is ultimately $O(\tau)$ as it follows from Theorem 3.1.



(a) (b)

FIG. 3.2. *Null-space projection method:* (a) *the relative norms of the true residual* $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ *of the projected system (solid lines) and the updated residual* $\bar{r}_k^{(x)}$ *(dashed lines)—the updated solution scheme* (3.4); *the relative norms of the errors* $\|x-\bar{x}_k\|_A/\|x-x_0\|_A$ *(solid lines) and* $\|y-\bar{y}_k\|_{B^TA^{-1}B}/\|y-\bar{y}_0\|_{B^TA^{-1}B}$ *(dashed lines)—the updated solution scheme* (3.4).

The residual $-B^T\bar{x}_k$ obviously does not depend on the back-substitution scheme; see Figure 3.3(d).

In contrast to the Schur complement reduction method, the inexactness is connected with the matrix $B$ instead of $A$. In practice, the sequential application of the matrix $(I-\Pi)A(I-\Pi)$ does not represent a symmetric operator. This is also reflected in the fact that we assume a general framework for computing the vector $x_k$ and analyze another projection of residuals $f-A\bar{x}_k-B\bar{y}_k$ and $\bar{r}_k^{(x)}$. Ideally at every iteration step we apply the matrix-vector product with the matrix $(I-\hat{\Pi})A(I-\hat{\Pi})$, where $\hat{\Pi}$ represents the orthogonal projector $\hat{\Pi}=(B+\Delta B)(B+\Delta B)^\dagger$ with $\|\Delta B\|\leq\tau\|B\|$. A question similar to one in subsection 2.1 arises as to whether we can apply the results of [22] directly to the system $(I-\hat{\Pi})A(I-\hat{\Pi})\hat{x}=(I-\hat{\Pi})f$. Theorem 3.1 shows that in finite precision arithmetic the residual $(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k$ will remain proportional to the parameter $\tau$. The theory of Greenbaum can be directly applied only if the multiplication by $(I-\Pi)A(I-\Pi)$ satisfies $\|\mathrm{fl}[(I-\Pi)A(I-\Pi)x]-(I-\Pi)A(I-\Pi)x\|\leq O(u)\|(I-\Pi)A(I-\Pi)\|\|x\|$ which is obviously not the case here. In the idealized case we have $\mathrm{fl}[(I-\Pi)A(I-\Pi)x]=(I-\hat{\Pi})A(I-\hat{\Pi})x$ and hence

$$\|\mathrm{fl}[(I-\Pi)A(I-\Pi)x]-(I-\Pi)A(I-\Pi)x\|\leq\frac{O(\tau)\kappa(B)}{1-\tau\kappa(B)}\|A\|\|x\|.$$

If we could improve this bound to satisfy $\|\mathrm{fl}[(I-\Pi)A(I-\Pi)x]-(I-\Pi)A(I-\Pi)x\|\leq\tau\|A\|\|x\|$, the outer iteration process could be viewed as an iteration in finite precision arithmetic with the roundoff unit equal to $\tau$ and the theory of Greenbaum would lead to the estimate

$$\|(I-\Pi)f-(I-\Pi)A(I-\Pi)\bar{x}_k-\bar{r}_k^{(x)}\|\leq\frac{O(\tau)}{1-\tau\kappa(B)}\|A\|(\|x\|+\bar{X}_k).$$

The numerical behavior of the null-space projection method was studied also in [1, 2], where the inner least squares are solved by the QR or LU factorization with $\tau=O(u)$ and the projected system is solved inexactly with the parameter $\eta$. Our Theorem 3.1 thus gives an answer to the question of how small the parameter $\eta$ can be in the outer iteration. Roughly speaking, when using the error or residual minimizing

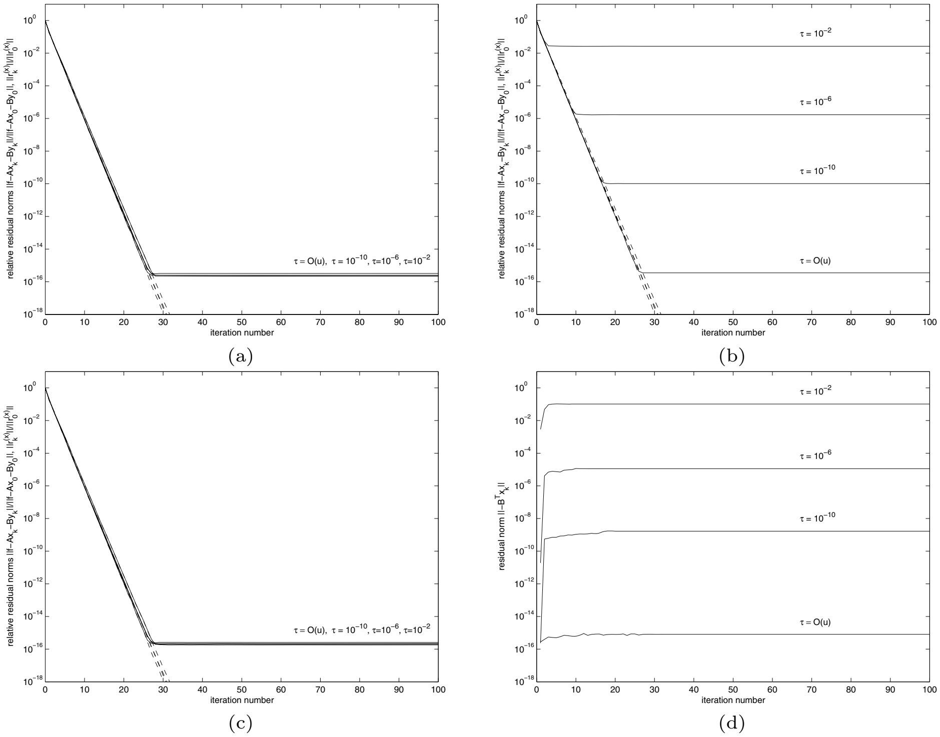


FIG. 3.3. *Null-space projection method: The relative norms of the true residual* $f - A\bar{x}_k - B\bar{y}_k$ *and the updated residual* $\bar{r}_k^{(x)}$ *(plots* (a), (b), *and* (c) *for the updated solution scheme* (3.4), *the direct substitution scheme* (3.5), *and the corrected direct substitution scheme* (3.6), *respectively);* (d) *the norms of the residual* $-B^T\bar{x}_k$*—the updated solution scheme* (3.4).

method for solving the projected Hessian system the backward error associated with the iterate $\bar{x}_k$ cannot be smaller than $O(u)\kappa(B)/[1 - O(u)\kappa(B)]$.

It is clear that no matter how we compute $\bar{x}_k$ and $\bar{y}_k$ we have the following relation between $(I - \Pi)f - (I - \Pi)A(I - \Pi)\bar{x}_k$, $f - A\bar{x}_k - B\bar{y}_k$, and $-B^T\bar{x}_k$:

$$(I - \Pi)f - (I - \Pi)A(I - \Pi)\bar{x}_k = (I - \Pi)(f - A\bar{x}_k - B\bar{y}_k) + (I - \Pi)A\Pi\bar{x}_k. \tag{3.15}$$

Owing to (3.8), $\Pi\bar{x}_k$ (and thus also $-B^T\bar{x}_k$) is $O(\tau)$. From Theorem 3.1 we have that $\|(I - \Pi)f - (I - \Pi)A(I - \Pi)\bar{x}_k\|$ is ultimately $O(\tau)$. Since $(I - \Pi)(f - A\bar{x}_k) = (I - \Pi)(f - A\bar{x}_k - B\bar{y}_k)$ for any $\bar{y}_k$ it also follows from Theorem 3.1 that the projection of $f - A\bar{x}_k - B\bar{y}_k$ onto $N(B^T)$ will ultimately reach $O(u)$. It is not clear from (3.15) whether the whole residual $f - A\bar{x}_k - B\bar{y}_k$ will be ultimately $O(\tau)$ or $O(u)$. It strongly depends on the back-substitution scheme used for computing the approximate solutions $y_{k+1}$. The following subsections show that the residual $f - A\bar{x}_k - B\bar{y}_k$ for the schemes with (3.4) (scheme A) and with (3.6) (scheme C) will finally reach $O(u)$, while the scheme B using (3.5) leads to the accuracy that is proportional only to $\tau$.

### 3.2. Scheme A: The updated approximate solution.

In this subsection we analyze the generic scheme with the update (3.4). This implementation does not require any additional solution of a least squares problem with the matrix $B$. Indeed, the computed direction vector $p_k^{(y)}$ is used to update both the iterate $y_k$ and the residual $\bar{r}_k^{(x)}$. As we will see, this algorithm computes the residual $f - A\bar{x}_k - B\bar{y}_k$

which will ultimately reach the level of roundoff unit $u$ independently of the fact that the inner least squares are solved with the accuracy determined by the parameter $\tau$.

THEOREM 3.2. *The gap between the residuals* $f - A\bar{x}_k - B\bar{y}_k$ *and* $\bar{r}_k^{(x)}$ *can be bounded as follows*:

$$\|f - A\bar{x}_k - B\bar{y}_k - \bar{r}_k^{(x)}\| \leq O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k),$$

*where* $\bar{Y}_k \equiv \max_{i=0,\dots,k} \|\bar{y}_i\|$. *The statement of the theorem remains true if we replace* $\bar{Y}_k$ *by* $\max\{\|y_0\|, \|p_i^{(y)}\|,\ i = 0, 1, \dots, k-1\}$.

*Proof.* The vector $\bar{x}_{k+1}$ satisfies (3.9) with $\|\Delta x_{k+1}\| \leq O(u)\bar{X}_{k+1}$, and similarly for $\bar{y}_{k+1}$ we have

$$\bar{y}_{k+1} = \bar{y}_k + \bar{p}_k^{(y)} + \Delta y_{k+1},\ \|\Delta y_{k+1}\| \leq u\|\bar{y}_k\| + (2u + u^2)\|\bar{p}_k^{(y)}\|$$

with $\|\Delta y_{k+1}\| \leq O(u)\bar{Y}_{k+1}$. The residual $\bar{r}_{k+1}^{(x)}$ satisfies (3.12) and thus $\|\Delta r_{k+1}^{(x)}\| \leq O(u)(\|\bar{r}_k^{(x)}\| + \|A\|\bar{X}_{k+1} + \|B\|\bar{Y}_{k+1})$. Using the above relations we obtain the recursive formula

$$f - A\bar{x}_{k+1} - B\bar{y}_{k+1} - \bar{r}_{k+1}^{(x)} = f - A\bar{x}_k - B\bar{y}_k - \bar{r}_k^{(x)} - A\Delta x_{k+1} - B\Delta y_{k+1} - \Delta r_{k+1}^{(x)}.$$

Taking the norms we get the following after some manipulation:

$$\|f - A\bar{x}_k - B\bar{y}_k - \bar{r}_k^{(x)}\| \leq O(u)\left(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k + \sum_{i=0}^{k-1}\|\bar{r}_i^{(x)}\|\right).$$

The statement can now be proved by induction on $k$. □

We have shown that $\bar{r}_k^{(x)}$ is a good approximation to $f - A\bar{x}_k - B\bar{y}_k$ independent of the fact that $\bar{p}_k^{(y)}$ are computed inexactly. Note that Theorem 3.1 can be derived using Theorem 3.2 due to $\|(I - \Pi)(f - A\bar{x}_k - \bar{r}_k^{(x)})\| = \|(I - \Pi)(f - A\bar{x}_k - B\bar{y}_k - \bar{r}_k^{(x)})\| \leq \|f - A\bar{x}_k - B\bar{y}_k - \bar{r}_k^{(x)}\|$. In Figure 3.3(a) we show the relative norms of $f - A\bar{x}_k - B\bar{y}_k$ (solid lines) and $\bar{r}_k^{(x)}$ (dashed lines). The results of our numerical experiment are in a good agreement with Theorem 3.2.

### 3.3. Scheme B: The approximate solution computed by a direct substitution.

In this subsection we analyze the scheme (3.5), which uses the directly computed right-hand side vector $f - Ax_k$. The computed $\bar{y}_k$ is then a solution of the perturbed problem

$$(B + \Delta B_k)\bar{y}_k \approx \mathrm{fl}(f - A\bar{x}_k) + \Delta c_k \tag{3.16}$$

with $\|\Delta B_k\| \leq \tau\|B\|$ and $\|\Delta c_k\| \leq \tau\|\mathrm{fl}(f - A\bar{x}_k)\|$. We will show that $(I - \Pi)\bar{r}_k^{(x)}$ is a good approximation of $f - A\bar{x}_k - B\bar{y}_k$ provided that both are above their level of maximum attainable accuracy.

THEOREM 3.3. *The gap between the residuals* $f - A\bar{x}_k - B\bar{y}_k$ *and* $(I - \Pi)\bar{r}_k^{(x)}$ *can be bounded by*

$$\begin{aligned}\|f - A\bar{x}_k - B\bar{y}_k - (I - \Pi)\bar{r}_k^{(x)}\| \leq{}& \tfrac{5\tau\kappa(B)}{1-\tau\kappa(B)}(\|f\| + \|A\|\|\bar{x}_k\|)\\ &+ O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k).\end{aligned}$$

*Proof.* Considering (3.16) it follows for the true residual that

$$\begin{aligned} f - A\bar{x}_k - B\bar{y}_k &= f - A\bar{x}_k - B(B+\Delta B_k)^\dagger[\mathrm{fl}(f - A\bar{x}_k) + \Delta c_k] \\ &= (I-\Pi)(f - A\bar{x}_k) + B[B^\dagger - (B+\Delta B_k)^\dagger]\mathrm{fl}(f - A\bar{x}_k) \\ &\quad + BB^\dagger[\mathrm{fl}(f - A\bar{x}_k) - (f - A\bar{x}_k)] - B(B+\Delta B_k)^\dagger\Delta c_k. \end{aligned}$$

Taking (3.16), the bounds on $B[B^\dagger - (B+\Delta B_k)^\dagger]$, $(B+\Delta B_k)^\dagger$, and Theorem 3.1 we get the desired result. ☐

When using the formula (3.5) the residual $f - A\bar{x}_k - B\bar{y}_k$ will not decrease below a level proportional to $\tau$, while $(I-\Pi)\bar{r}_k^{(x)}$ converges beyond the level $O(u)$. This result is illustrated by our numerical experiment. In Figure 3.3(b) we plotted the relative norms of $f - A\bar{x}_k - B\bar{y}_k$ (solid lines) and $\bar{r}_k^{(x)}$ (dashed lines).

**3.4. Scheme C: The approximate solution computed with a corrected direct substitution.** In this subsection we analyze the scheme (3.6) requiring a solution of two least squares problems with $B$. We show that its behavior is similar to the algorithm using the update (3.4). We prove that under certain assumptions the true residual $f - A\bar{x}_k - B\bar{y}_k$ converges ultimately to the $O(u)$ level. The difference is that while Theorem 3.2 holds without any additional conditions, here we have a situation analogous to the behavior of nonstationary iterative methods (see [23, Chapter 16]).

THEOREM 3.4. *Provided that for sufficiently large step $k$ the computed vector $\bar{x}_k$ stagnates, i.e., we have $\|\bar{x}_{k+1} - \bar{x}_k\| \le O(u)\bar{X}_{k+1}$, there exists some iteration step $k_0$ such that*

$$\|f - A\bar{x}_k - B\bar{y}_k - (I-\Pi)\bar{r}_k^{(x)}\| \le O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k) \tag{3.17}$$

*holds for all $k \ge k_0$.*

*Proof.* The vector $\bar{y}_{k+1}$ satisfies $\bar{y}_{k+1} = \bar{y}_k + \bar{q}_k^{(y)} + \Delta y_{k+1}$ and $\|\Delta y_{k+1}\| \le O(u)\bar{Y}_{k+1}$, where $\bar{q}_k^{(y)}$ is the solution of the problem $(B+\Delta B_k)\bar{q}_k^{(y)} \approx \mathrm{fl}(f - A\bar{x}_{k+1} - B\bar{y}_k) + \Delta c_k$ with $\|\Delta B_k\| \le \tau\|B\|$ and $\|\Delta c_k\| \le \tau\|\mathrm{fl}(f - A\bar{x}_{k+1} - B\bar{y}_k)\|$. For $f - A\bar{x}_{k+1} - B\bar{y}_{k+1}$ we can then write

$$\begin{aligned} f - A\bar{x}_{k+1} - B\bar{y}_{k+1} &= (I-\Pi)(f - A\bar{x}_{k+1}) + G_k(f - A\bar{x}_{k+1} - B\bar{y}_k) \\ &\quad - B(B+\Delta B_k)^\dagger\Delta c_k + h_k, \end{aligned}$$

where $G_k = B[B^\dagger - (B+\Delta B_k)^\dagger]$ and $h_k = -B(B+\Delta B_k)^\dagger[\mathrm{fl}(f - A\bar{x}_{k+1} - B\bar{y}_k) - (f - A\bar{x}_{k+1} - B\bar{y}_k)] - B\Delta y_{k+1}$. Projecting $f - A\bar{x}_{k+1} - B\bar{y}_{k+1}$ onto $R(B)$ and taking norms, we obtain

$$\begin{aligned} \|\Pi(f - A\bar{x}_{k+1} - B\bar{y}_{k+1})\| &\le \left[\|G_k\| + \tau\|B(B+\Delta B_k)^\dagger\|\right]\|f - A\bar{x}_{k+1} - B\bar{y}_k\| \\ &\quad + \tau\|B(B+\Delta B_k)^\dagger\|\|\mathrm{fl}(f - A\bar{x}_{k+1} - B\bar{y}_k) - (f - A\bar{x}_{k+1} - B\bar{y}_k)\| + \|h_k\|. \end{aligned}$$

The term $\|f - A\bar{x}_{k+1} - B\bar{y}_k\|$ can be further bounded by

$$\|f - A\bar{x}_{k+1} - B\bar{y}_k\| \le \|(I-\Pi)(f - A\bar{x}_{k+1})\| + \|\Pi(f - A\bar{x}_k - B\bar{y}_k)\| + \|A(\bar{x}_{k+1} - \bar{x}_k)\|$$

which together with the bound on $\|G_k\|$, $\|h_k\| \le O(u)(\|f\| + \|A\|\bar{X}_{k+1} + \|B\|\bar{Y}_{k+1})$, and $\tau\|B(B+\Delta B_k)^\dagger\| \le \tau\kappa(B)[1-\tau\kappa(B)]^{-1} < 1$ leads to

$$\begin{aligned}
&\|\Pi(f - A\bar{x}_{k+1} - B\bar{y}_{k+1})\| \\
&\quad \le \frac{3\tau\kappa(B)}{1-\tau\kappa(B)} \left[\|\Pi(f - A\bar{x}_k - B\bar{y}_k)\| + \|(I-\Pi)(f - A\bar{x}_{k+1})\| + \|A\|\|\bar{x}_{k+1} - \bar{x}_k\|\right] \\
&\qquad + O(u)(\|f\| + \|A\|\bar{X}_{k+1} + \|B\|\bar{Y}_{k+1}).
\end{aligned}$$

After the recursive use of the previous inequality we obtain

$$\begin{aligned}
(3.18)\qquad \|\Pi(f - A\bar{x}_k - B\bar{y}_k)\| &\le \left(\frac{3\tau\kappa(B)}{1-\tau\kappa(B)}\right)^k \|f - A\bar{x}_0 - B\bar{y}_0\| \\
&+ \sum_{i=0}^{k-1} \left(\frac{3\tau\kappa(B)}{1-\tau\kappa(B)}\right)^{k-i} \left[\|(I-\Pi)(f - A\bar{x}_{i+1})\| + \|A\|\|\bar{x}_{i+1} - \bar{x}_i\|\right] \\
&+ O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k).
\end{aligned}$$

Under the assumption on the stagnation of iterates there exist some index $k_0$ such that the second term on the right-hand side of (3.18) will be of order $O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k)$ for all iteration steps $k \ge k_0$. Finally, from Theorem 3.2 we have $\|(I-\Pi)(f - A\bar{x}_k) - (I-\Pi)\bar{r}_k^{(x)}\| \le O(u)(\|f\| + \|A\|\bar{X}_k + \|B\|\bar{Y}_k)$. ☐

Theorem 3.4 shows that $f - A\bar{x}_k - B\bar{y}_k$ will ultimately reach the $O(u)$ level. As soon as the approximate solutions $\bar{x}_k$ stagnate with $\|\bar{x}_{k+1} - \bar{x}_k\| \le O(u)\bar{X}_{k+1}$, the rate of convergence of this process is roughly given by the factor $3\tau\kappa(B)[1-\tau\kappa(B)]^{-1}$. Note that similar to subsection 2.4 the assumption on the stagnation is not restrictive. The numerical results on a model example are shown in Figure 3.3(c), which reports the relative norms of $f - A\bar{x}_k - B\bar{y}_k$ (solid lines) and $\bar{r}_k^{(x)}$ (dashed lines), and are in good agreement with Theorem 3.4.

### 3.5. Forward error analysis.

In this subsection we look at the maximum attainable accuracy measured by errors $x - \bar{x}_k$ and $y - \bar{y}_k$. The analysis is very similar to the Schur complement reduction method and therefore we focus only on issues particular to the null-space projection method. We recall that relation (2.25) gives the universal bounds (2.26), (2.27), and (2.28). Independent of the back-substitution scheme used for computing $\bar{y}_k$, the terms $\gamma_2\| - B^T\bar{x}_k\|$ and $\gamma_3\| - B^T\bar{x}_k\|$ on the right-hand side of (2.26) and (2.27), respectively, are always proportional to $\tau$. The terms with $f - A\bar{x}_k - B\bar{y}_k$ depend on the back-substitution formula and their final magnitude will be at most $O(\tau)$, leading to similar conclusions on errors as in subsection 2.5. The estimate for $\|x - \bar{x}_k\|_A$ is given in the following theorem.

Theorem 3.5. *The A-norm of the error $x - \bar{x}_k$ can be bounded as*

$$(3.19)\qquad \|x - \bar{x}_k\|_A \le \delta_1\| - B^T\bar{x}_k\| + \delta_2\|(I-\Pi)(f - A\bar{x}_k)\|,$$

*where $\delta_1 \equiv \|A\|^{1/2}/\sigma_{min}(B)$ and $\delta_2 \equiv \sigma_{min}^{-1/2}(A)$ are constants independent of the iteration step $k$.*

*Proof.* Since $(I-\Pi)A(x - \bar{x}_k) = (I-\Pi)(f - A\bar{x}_k)$, $B^Tx = 0$ and $\|B(B^TB)^{-1}\| = \sigma_{min}^{-1}(B)$, $\|x - \bar{x}_k\|_A^2$ can be written as

$$\begin{aligned}
(3.20)\quad \|x - \bar{x}_k\|_A^2 &= (\Pi(x - \bar{x}_k), A(x - \bar{x}_k)) + ((I-\Pi)A(x - \bar{x}_k), x - \bar{x}_k) \\
&\le \|A^{1/2}\|\|x - \bar{x}_k\|_A(\|B(B^TB)^{-1}\|\|B^T(x - \bar{x}_k)\| + \|(I-\Pi)(f - A\bar{x}_k)\|).
\end{aligned}$$

Dividing both sides by $\|x - \bar{x}_k\|_A$ gives the statement (3.19). $\square$

The first term on the right-hand side of (3.19) should be zero in exact arithmetic. The computed $\bar{x}_k$, however, does not fulfill $-B^T\bar{x}_k = 0$ and its departure from $N(B^T)$ was discussed in (3.8). The second term converges to zero in exact arithmetic and it is related to the projected residual $(I - \Pi)(f - A\bar{x}_k)$; see Theorem 3.14. The result for $y - \bar{y}_k$ can be obtained from (3.19) using (2.28). Provided that $\bar{r}_k^{(x)}$ is larger than $O(\tau)$, $\|x - \bar{x}_k\|_A$ is then well approximated by $\delta_2\|(I - \Pi)\bar{r}_k^{(x)}\|$.

**4. Conclusions.** In this paper we have looked at the numerical behavior of certain inexact saddle point solvers. In particular, for several mathematically equivalent implementations we studied the influence of inexact solving of the inner systems and estimated their maximum attainable accuracy. When considering the outer iteration process our rounding error analysis has led to results similar to ones which can be obtained assuming exact arithmetic. The situation was different when we looked at the residuals in the saddle point system. We have shown that some implementations lead ultimately to residuals on the roundoff unit level independently of the fact that the inner systems were solved inexactly on a much higher level $\tau$. Indeed, our results confirmed that the generic and actually the cheapest implementations deliver the approximate solutions which satisfy either the second or the first block equation to the working accuracy. In addition, the schemes with the corrected direct substitution are also very attractive. We gave a theoretical explanation for the behavior which was probably observed or is already tacitly known. The implementations that we pointed out as optimal are actually those which are widely used and suggested in applications. It appears that, when measured in terms of the errors, the maximum attainable accuracy level is similar for all considered implementations and is proportional to the parameter which measures the inexactness in solving the inner systems.

**Acknowledgments.** The authors would like to thank the anonymous referees for their valuable comments and suggestions and Volker Mehrmann for his discussion during the MAT-TRIAD 2005 meeting, which actually stimulated this work.

## REFERENCES

[1] M. Arioli, *The use of QR factorization in sparse quadratic programming and backward error issues*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 825–839.

[2] M. Arioli and L. Baldini, *A backward error analysis of a null space algorithm in sparse quadratic programming*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 425–442.

[3] J. Atanga and D. Silvester, *Iterative methods for stabilized mixed velocity-pressure finite elements*, Internat. J. Numer. Methods Fluids, 14 (1992), pp. 71–81.

[4] C. Bacuta, *A unified approach for Uzawa algorithms*, SIAM J. Numer. Anal., 44 (2006), pp. 2633–2649.

[5] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[6] A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[7] A. Bouras and V. Frayssé, *Inexact matrix-vector products in Krylov methods for solving linear systems: A relaxation strategy*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 660–678.

[8] A. Bouras, V. Frayssé, and L. Giraud, *A Relaxation Strategy for Inner-Outer Linear Solvers in Domain Decomposition Methods*, Technical report TR/PA/00/17, CERFACS, France, 2000.

[9] D. Braess, P. Deuflhard, and K. Lipnikov, *A subspace cascadic multigrid method for mortar elements*, Computing, 69 (2002), pp. 205–225.

[10] D. Braess and R. Sarazin, *An efficient smoother for the Stokes problem*, Appl. Numer. Math., 23 (1997), pp. 3–19.

[11] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[12] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Inexact Uzawa algorithms for nonsymmetric saddle point problems*, Math. Comp., 69 (2000), pp. 667–689.

[13] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Ser. Comput. Math. 15, Springer-Verlag New York, 1991.

[14] J. W. Demmel, N. J. Higham, and R. S. Schreiber, Stability of *block LU factorization*, Numer. Linear Algebra Appl., 2 (1995), pp. 173–190.

[15] H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[16] A. Frommer and D. B. Szyld, *H-splittings and two-stage iterative methods*, Numer. Math., 63 (1992), pp. 345–356.

[17] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press Inc., London, 1981.

[18] L. Giraud, S. Gratton, and J. Langou, *Convergence in backward error of relaxed GMRES*, SIAM J. Sci. Comput., 29 (2007), pp. 710–728.

[19] G. H. Golub and Q. Ye, *Inexact preconditioned conjugate gradient method with inner-outer iteration*, SIAM J. Sci. Comput., 21 (1999), pp. 1305–1320.

[20] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.

[21] A. Greenbaum, *Accuracy of computed solutions from conjugate-gradient-like methods*, in Advances in Numerical Methods for Large Sparse Sets of Linear Systems, vol. 10, M. Natori and T. Nodera, eds., Keio University, Yokohama, Japan, 1994, pp. 126–138.

[22] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 535–551.

[23] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[24] P. Jiránek and M. Rozložník, *Limiting accuracy of segregated solution methods for nonsymmetric saddle point problems*, J. Comput. Appl. Math., to appear.

[25] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[26] J. Maryška, M. Rozložník, and M. Tůma, *Schur complement reduction in the mixed-hybrid approximation of Darcy's law: Rounding error analysis*, J. Comput. Appl. Math., 117 (2000), pp. 159–173.

[27] N. K. Nichols, *On the convergence of two-stage iterative processes for solving linear equations*, SIAM J. Numer. Anal., 10 (1973), pp. 460–469.

[28] I. Perugia and V. Simoncini, *Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations*, Numer. Linear Algebra Appl., 7 (2000), pp. 585–616.

[29] A. Ramage and A. J. Wathen, *Iterative solution techniques for the Stokes and Navier-Stokes equations*, Internat. J. Numer. Methods Fluids, 19 (1994), pp. 67–83.

[30] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 368–391.

[31] V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.

[32] Z. Strakoš and P. Tichý, *On error estimation in the conjugate gradient method and why it works in finite precision computations*, Electron. Trans. Numer. Anal., 13 (2002), pp. 56–80.

[33] J. van den Eshof and G. L. G. Sleijpen, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.

[34] P. A. Wedin, *Perturbation theory for pseudo-inverses*, BIT, 13 (1973), pp. 217–232.

[35] C. Wieners and B. I. Wohlmuth, *Duality estimates and multigrid analysis for saddle point problems arising from mortar discretizations*, SIAM J. Sci. Comput., 24 (2003), pp. 2163–2184.

[36] W. Zulehner, *A class of smoothers for saddle point problems*, Computing, 65 (2000), pp. 227–246.

[37] W. Zulehner, *Analysis of iterative methods for saddle point problems: A unified approach*, Math. Comp., 71 (2002), pp. 479–505.

SIAM J. MATRIX ANAL. APPL. © 2008 Society for Industrial and Applied Mathematics
Vol. 29, No. 4, pp. 1322–1342

# NEW FAST AND ACCURATE JACOBI SVD ALGORITHM. I*

ZLATKO DRMAČ† AND KREŠIMIR VESELIĆ‡

*Dedicated to the memory of Patricia J. Eberlein, whose enthusiasm and belief in the powers of the Jacobi methods were a constant inspiration*

**Abstract.** This paper is the result of concerted efforts to break the barrier between numerical accuracy and run-time efficiency in computing the fundamental decomposition of numerical linear algebra—the singular value decomposition (SVD) of general dense matrices. It is an unfortunate fact that the numerically most accurate one-sided Jacobi SVD algorithm is several times slower than generally less accurate bidiagonalization-based methods such as the QR or the divide-and-conquer algorithm. Our quest for a highly accurate and efficient SVD algorithm has led us to a new, superior variant of the Jacobi algorithm. The new algorithm has inherited all good high accuracy properties of the Jacobi algorithm, and it can outperform the QR algorithm.

**Key words.** Jacobi method, singular value decomposition, eigenvalues

**AMS subject classifications.** 15A09, 15A12, 15A18, 15A23, 65F15, 65F22, 65F35

**DOI.** 10.1137/050639193

## 1. Introduction.

In 1846, Jacobi [25] introduced a new simple and accurate algorithm for diagonalization of symmetric matrices. The algorithm starts with symmetric matrix $H^{(0)} = H \in \mathbb{R}^{n\times n}$ and then generates a sequence of congruences, $H^{(k+1)} = (V^{(k)})^T H^{(k)} V^{(k)}$, where $V^{(k)}$ is plane rotation; i.e., $V^{(k)}$ differs from the identity only at the cleverly chosen positions $(p_k, p_k)$, $(p_k, q_k)$, $(q_k, p_k)$, $(q_k, q_k)$, where

$$\begin{pmatrix} V^{(k)}_{p_k,p_k} & V^{(k)}_{p_k,q_k} \\ V^{(k)}_{q_k,p_k} & V^{(k)}_{q_k,q_k} \end{pmatrix} = \begin{pmatrix} \cos\phi_k & \sin\phi_k \\ -\sin\phi_k & \cos\phi_k \end{pmatrix}.$$

The angle $\phi_k$ is determined to annihilate the $(p_k, q_k)$ and $(q_k, p_k)$ positions in $H^{(k)}$. Simple trigonometry reveals that in the nontrivial case ($H^{(k)}_{p_kq_k} \neq 0$) we can take

$$\cot 2\phi_k = \frac{H^{(k)}_{q_kq_k} - H^{(k)}_{p_kp_k}}{2H^{(k)}_{p_kq_k}} \quad\text{and}\quad \tan\phi_k = \frac{\operatorname{sign}(\cot 2\phi_k)}{|\cot 2\phi_k| + \sqrt{1+\cot^2 2\phi_k}} \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right],$$

where $\phi_k$ is the smaller of two angles satisfying the requirements. (If $H^{(k)}_{p_kq_k} = 0$, then $V^{(k)} = I$, the identity.) Under suitable pivot strategies $k \mapsto (p_k, q_k)$, the sequence $(H^{(k)})_{k=0}^{\infty}$ converges to diagonal matrix $\Lambda$, and the product $V^{(0)}V^{(1)}\cdots V^{(k)}\cdots$ converges to the orthogonal matrix $V$ of eigenvectors of $H$, $HV = V\Lambda$. The convergence is monitored using the off-norm, $\Omega(H) = \sqrt{\sum_{i\neq j} H_{ij}^2}$, for which one easily shows the monotonicity $\Omega^2(H^{(k+1)}) = \Omega^2(H^{(k)}) - 2(H^{(k)})^2_{p_k,q_k} \leq \Omega^2(H^{(k)})$.

---

[*] Received by the editors August 20, 2005; accepted for publication (in revised form) by M. Chu June 5, 2007; published electronically January 4, 2008. This work was supported by the Volkswagen–Stiftung grant *Designing Highly Accurate Algorithms for Eigenvalue and Singular Value Decompositions.*

http://www.siam.org/journals/simax/29-4/63919.html

[†] Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia (drmac@math.hr). The work of this author was supported by the Croatian Ministry of Science and Technology under grant 0037120 (*Numerical Analysis and Matrix Theory*).

[‡] Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Postfach 940, D-58084 Hagen, Germany (kresimir.veselic@FernUni-Hagen.de).

Hestenes [23] noted that the Jacobi method can be used to compute the SVD of general matrices. If $A$ is of full column rank[1] and if we define $H = A^T A$, then the application of the method to $H$, $H^{(k+1)} = (V^{(k)})^T H^{(k)} V^{(k)}$, can be represented by the sequence $A^{(k+1)} = A^{(k)} V^{(k)}$. To determine the parameters of $V^{(k)}$ we only need the four pivot elements of $H^{(k)}$, that is, the $2 \times 2$ Gram matrix of the $p_k$th and the $q_k$th column of $A^{(k)}$. The limit matrix of $(A^{(k)})_{k=0}^{\infty}$ is $U\Sigma$, where the columns of orthonormal $U$ are the left singular vectors and the diagonal matrix $\Sigma$ carries the singular values along its diagonal. The accumulated product of Jacobi rotations is orthogonal matrix $V$ of the eigenvectors of $H$. The SVD of $A$ is $A = U\Sigma V^T$.

The development of fast methods based on reduction of $A$ to bidiagonal form shifted interest away from the Jacobi SVD method—bidiagonalization-based routines xGESVD [11] and xGESDD [21] from LAPACK [1] can in some cases be ten or even fifteen times faster than the Jacobi SVD. However, Demmel and Veselić [12] showed that the Jacobi algorithm is more accurate than any other algorithm that first reduces the matrix to bidiagonal form. Some classes of matrices that appear ill-conditioned with respect to SVD computation in fact define the SVD perfectly well, and the ill-conditioning is artificial. For instance, if $A \in \mathbb{R}^{m\times n}$ is such that $\min_{D=\mathrm{diag}} \kappa_2(AD)$ is moderate, then the Jacobi SVD algorithm computes all singular values $\sigma_1 \geq \cdots \geq \sigma_n$ with guaranteed number of correct digits independent of the size of $\kappa_2(A) = \sigma_{\max}(A)/\sigma_{\min}(A) \equiv \sigma_1/\sigma_n$. The Jacobi method correctly deals with artificial ill-conditioning (e.g., grading), while the bidiagonalization-based methods do not.

In this paper we present a new preconditioned Jacobi SVD algorithm which provides higher accuracy with efficiency comparable to the bidiagonalization-based methods. In section 2 we show that rank revealing QR factorization can be used as an efficient preconditioner for the Jacobi SVD algorithm, and we reduce the problem to SVD computation of structured triangular matrices. Important detail of choosing $A$ or $A^T$ as input to the new algorithm is discussed in section 3. The dilemma "$A$ or $A^T$" generates interesting mathematical questions leading us to study entropy of the set of normalized diagonals of the adjoint orbit of a positive definite matrix. The basic structure of the new Jacobi SVD algorithm is developed in section 4. Numerical properties are analyzed in section 5. It is shown that the backward perturbation has structure which allows scaling invariance of the condition number for the forward error. Implementation details of a new Jacobi SVD for triangular matrices and results of numerical testing of the preconditioned Jacobi SVD are given in the second part of this paper [18], where we demonstrate the potential of the new approach.

## 2. QR factorization as preconditioner for the Jacobi SVD algorithm.

In the case $m \gg n$, the QR factorization $A = Q \begin{pmatrix} R^T & 0 \end{pmatrix}^T$ is an attractive preprocessor for the Jacobi SVD algorithm, because Jacobi rotations can be applied to $R \in \mathbb{R}^{n\times n}$ instead of $A \in \mathbb{R}^{m\times n}$. If the QR factorization is computed with a rank revealing column pivoting, $A\Pi = Q \begin{pmatrix} R^T & 0 \end{pmatrix}^T$, then the additional structure of $R$ opens quite a few possibilities for more efficient SVD computation by the Jacobi algorithm.

### 2.1. Faster convergence.

Veselić and Hari [34] noted that the eigenvalues of symmetric positive definite $H$ can be computed more efficiently if the Jacobi SVD method is applied to the lower triangular factor $L$ from the pivoted Cholesky factorization $P^T H P = LL^T$. If $H = A^T A$, $\mathrm{rank}(A) = n$, then Cholesky factorization with pivoting of $H$ corresponds to the QR factorization with column pivoting

---

[1] This is only a temporary assumption for the sake of simplicity.

$A\Pi = Q\begin{pmatrix} R^T & 0 \end{pmatrix}^T$, $L = R^T$. Hence, the Jacobi SVD on $R^T$ should have better convergence than if applied to $R$, and the preconditioning is performed simply by taking $R^T$ instead of $R$ as input. Implicitly, this is one step of the Rutishauser [31] LR diagonalization method $R^TR \rightsquigarrow RR^T$, which has a nontrivial diagonalizing effect.

Let $R = \left(\begin{smallmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{smallmatrix}\right)$, where the diagonal blocks are $k \times k$ and $(n-k) \times (n-k)$, and consider the corresponding block partitions of $H = R^TR$ and $M = RR^T$:

$$H = \begin{pmatrix} H_{[11]} & H_{[12]} \\ H_{[21]} & H_{[22]} \end{pmatrix} = \left(\begin{array}{c|c} R_{[11]}^T R_{[11]} & R_{[11]}^T R_{[12]} \\ \hline R_{[12]}^T R_{[11]} & R_{[12]}^T R_{[12]} + R_{[22]}^T R_{[22]} \end{array}\right),$$

$$M = \begin{pmatrix} M_{[11]} & M_{[12]} \\ M_{[21]} & M_{[22]} \end{pmatrix} = \left(\begin{array}{c|c} R_{[11]} R_{[11]}^T + R_{[12]} R_{[12]}^T & R_{[12]} R_{[22]}^T \\ \hline R_{[22]} R_{[12]}^T & R_{[22]} R_{[22]}^T \end{array}\right). \tag{2.1}$$

Note that the $(1,1)$ block in $M$ is increased, and the $(2,2)$ block is decreased, i.e., $\mathrm{Trace}(M_{[11]}) = \mathrm{Trace}(H_{[11]}) + \|R_{[12]}\|_F^2$, $\mathrm{Trace}(M_{[22]}) = \mathrm{Trace}(H_{[22]}) - \|R_{[12]}\|_F^2$.

This redistribution of the mass of the diagonal blocks makes the gap between the dominant and subdominant parts of the spectrum more visible on the diagonal. Using the monotonicity of the spectrum, we also conclude that properly ordered eigenvalues $\lambda_i(\cdot)$ of the diagonal blocks satisfy $\lambda_i(M_{[11]}) \geq \lambda_i(H_{[11]})$, $\lambda_j(M_{[22]}) \leq \lambda_j(H_{[22]})$, $1 \leq i \leq k$, $1 \leq j \leq n-k$. With suitable pivot strategy, this has positive impact on the structure of Jacobi rotations (smaller angles) in the off-diagonal blocks in (2.1). Moreover, the off-diagonal blocks of $M$ are expected to be smaller than those in $H$.

PROPOSITION 2.1. *With the partition of* $R$ *and* (2.1) *define* $R_{[1:]} = \begin{pmatrix} R_{[11]} & R_{[12]} \end{pmatrix}$, $R_{[:2]} = \left(\begin{smallmatrix} R_{[12]} \\ R_{[22]} \end{smallmatrix}\right)$, $\cos\Phi = H_{[11]}^{-1/2} H_{[12]} H_{[22]}^{-1/2}$, *and* $\cos\Psi = M_{[11]}^{-1/2} M_{[12]} M_{[22]}^{-1/2}$. *Then, with* $\|\cdot\| \in \{\|\cdot\|_2 \equiv \sigma_{\max}(\cdot), \|\cdot\|_F\}$,

$$\text{(i)} \quad \|M_{[12]}\| \leq \frac{\sigma_{\max}(R_{[22]})}{\sigma_{\min}(R_{[11]})}\|H_{[12]}\|\ ; \quad \text{(ii)} \quad \|\cos\Psi\| \leq \frac{\sigma_{\max}(R_{[:2]})}{\sigma_{\min}(R_{[1:]})}\|\cos\Phi\|. \tag{2.2}$$

*Proof.* To prove the well-known relation (i), one notes that $M_{[12]} = R_{[11]}^{-T} H_{[12]} R_{[22]}^T$. For the new estimate (ii), we use the connections between the LQ factorization, Cholesky factorization and the positive definite matrix square root to conclude that there exist orthogonal matrices $S_1$, $S_2$ such that

$$\cos\Psi = (R_{[1:]}R_{[1:]}^T)^{-1/2}\ S_1\ \cos\Phi\ (R_{[:2]}^T R_{[:2]})^{1/2}\ S_2.$$

Taking the norm completes the proof. □

Let $\xi = \xi(R,k) = \sigma_{\max}(R_{[22]})/\sigma_{\min}(R_{[11]})$. If $\xi < 1$, then $\|M_{[12]}\|_F \leq \xi\|H_{[12]}\|_F < \|H_{[12]}\|_F$. Thus, a smaller value of $\xi$ implies more block diagonal structure in $M$ than in $H$. Now, it is the task of the rank revealing pivoting in the QR factorization to find index $k$ for which $\xi \ll 1$. If the pivoting is done right, and if the singular values of $R$ are distributed so that $\sigma_k \gg \sigma_{k+1}$ for some $k$, then $\xi$ will be much smaller than one. See [7], [8] for a detailed analysis related to (2.2(i)).

Note that (2.2(ii)) estimates scaled off-diagonal blocks, which is relevant for the convergence of the Jacobi algorithm. Relevant separation is given by $\zeta = \zeta(R,k) = \sigma_{\max}(R_{[:2]})/\sigma_{\min}(R_{[1:]})$. In general, $\xi < 1$ does not imply $\zeta < 1$, but an additional factorization will provide that stronger separation.

THEOREM 2.2. *Let* $R = LQ_L$ *be the LQ factorization of* $R$,

$$\begin{pmatrix} R_{[11]} & R_{[12]} \\ 0 & R_{[22]} \end{pmatrix} = \begin{pmatrix} L_{[11]} & 0 \\ L_{[21]} & L_{[22]} \end{pmatrix} Q_L = LQ_L.$$

*Then we have the following monotonicity relations*:

$$\sigma_i\left(\begin{pmatrix} L_{[11]} \\ L_{[21]} \end{pmatrix}\right) \geq \sigma_i(L_{[11]}) = \sigma_i(R_{[1:]}) \geq \sigma_i(R_{[11]}),\quad i = 1,\ldots,k; \tag{2.3}$$

$$\sigma_j(L_{[22]}) \leq \sigma_j(\begin{pmatrix} L_{[21]} & L_{[22]} \end{pmatrix}) = \sigma_j(R_{[22]}) \leq \sigma_j(R_{[:2]}),\quad j = 1,\ldots,n-k. \tag{2.4}$$

*In particular, if* $\sigma_{\min}(R_{[1:]}) > \sigma_{\max}(R_{[22]})$, *then* $\xi(L^T,k) < 1$, $\zeta(L^T,k) < 1$, *and* $\|L_{[21]}\| < \|R_{[12]}\|$. *In that case,* $M = RR^T$ *is a shifted quasi-definite matrix.*

*Proof.* The inequalities (2.3) and (2.4) are obtained by an application of the monotonicity principle to the corresponding eigenvalues. Combination of the inequalities with the assumption $\sigma_{\min}(R_{[1:]}) > \sigma_{\max}(R_{[22]})$ gives the bounds on $\xi$ and $\zeta$. Since $L_{[21]} = R_{[22]}R_{[12]}^T L_{[11]}^{-T}$ we have, for $\|\cdot\| \in \{\|\cdot\|_2, \|\cdot\|_F\}$, $\|L_{[21]}\| \leq \frac{\sigma_{\max}(R_{[22]})}{\sigma_{\min}(R_{[1:]})}\|R_{[12]}\| < \|R_{[12]}\|$. Finally, note that $M - s^2 I$ is quasi-definite for any $s \in (\sigma_{\max}(R_{[22]}), \sigma_{\min}(R_{[1:]}))$. $\square$

*Remark* 2.1. Theorem 2.2 introduces gap $\gamma(R,k) = \sigma_{\max}(R_{[22]})/\sigma_{\min}(R_{[1:]})$, which from the pivoted QR factorization requires less than the condition $\sigma_{\max}(R_{[22]}) < \sigma_{\min}(R_{[11]})$. If $\gamma < 1$, then the off-diagonal block in $M^{(1)} = L^T L$ satisfies $\|M^{(1)}_{[21]}\| \leq (\sigma_{\max}(L_{[22]})/\sigma_{\min}(L_{[11]}))(\sigma_{\max}(R_{[22]})/\sigma_{\min}(R_{[11]}))\|H_{21}\|$.

Now, it is reasonable to expect that the one-sided Jacobi SVD algorithm applied to $L$ runs quite differently than if applied to the initial $A$. The structure of $L$ calls for modifications, and the result is a new Jacobi-type algorithm designed for triangular matrices. Its complete description is in the second part of this report [18].

*Remark* 2.2. It is well known that repeated application of the step "do QR factorization and transpose $R$" is actually a simple way to approximate the SVD; see [29], [20], [32]. Fernando and Parlett [20] first realized that "the use of a preconditioner for cyclic Jacobi is not a futile effort." Here we stress the term *preconditioner* and use of the implicit Cholesky SVD as preconditioner for Jacobi iterations.

**2.2. Rank deficient cases.** Consider the case of $A$ with low numerical rank. The task is to compute the singular values with standard error bound. In a rank revealing QR factorization, the computed $\tilde{R}$, $\tilde{Q}$ satisfy backward stability relations

$$(A + \delta A)\Pi = \hat{Q}\begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix},\quad \|\delta A(:,i)\|_2 \leq \varepsilon_{qr}\|A(:,i)\|_2,\quad i = 1,\ldots,n;\quad \|\tilde{Q} - \hat{Q}\|_F \leq \varepsilon_{qr},$$

where $\hat{Q}$ is orthogonal and $\varepsilon_{qr}$ is bounded by a moderate $f(m,n)$ times the round-off $\boldsymbol{\varepsilon}$. Suppose there is an index $k \in \{1,\ldots,n\}$ such that $\tilde{R}$ can be partitioned as

$$\tilde{R} = \begin{pmatrix} \tilde{R}_{[11]} & \tilde{R}_{[12]} \\ 0 & \tilde{R}_{[22]} \end{pmatrix},\quad \text{where } \tilde{R}_{[22]} \in \mathbb{R}^{(n-k)\times(n-k)} \text{ is sufficiently small.} \tag{2.5}$$

If we set $\tilde{R}_{[22]}$ to zero, then we will implicitly continue working with the matrix

$$\hat{Q}\begin{pmatrix} \tilde{R}_{[11]} & \tilde{R}_{[12]} \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \left(A + \delta A - \hat{Q}\begin{pmatrix} 0 & 0 \\ 0 & \tilde{R}_{[22]} \\ 0 & 0 \end{pmatrix}\Pi^T\right)\Pi \equiv (A + \Delta A)\Pi. \tag{2.6}$$

If $\|\tilde{R}_{[22]}\|_2/\|A\|_2$ is of the order of $\varepsilon_{qr}$, then replacing $\tilde{R}_{[22]}$ with zero is in the matrix norm a backward stable operation for singular value computation with classical error bound—the perturbation in each singular value is small compared to $\|A\|_2 = \sigma_{\max}(A)$.

Further, (2.6) is the QR factorization of $(A+\Delta A)\Pi$, where for the computed orthogonal factor we can keep $\tilde{Q}$. If we compute the LQ factorization of the $k \times n$ matrix, $\hat{R} \equiv (\,\tilde{R}_{[11]} \quad \tilde{R}_{[12]}\,) = LQ_1$, then the Jacobi iterations work on substantially smaller $k \times k$ lower triangular matrix $L$. Thus, a rank revealing ULV decomposition is in this case an excellent preprocessor for the Jacobi SVD algorithm.

Theorem 2.3. *Let the block $\tilde{R}_{[11]}$ in* (2.6) *be nonsingular and let $\tilde{R}_{[11]} = DT$, where $D$ is diagonal with $\sigma_{\min}(D) = |\tilde{r}_{kk}|$. Further, let the truncated block $\tilde{R}_{[22]}$ satisfy $\|\tilde{R}_{[22]}\|_2 \le \tau|\tilde{r}_{kk}|$. If $\tilde{\sigma}_1 \ge \cdots \ge \tilde{\sigma}_k$ and $\hat{\sigma}_1 \ge \cdots \ge \hat{\sigma}_k$ are the nonzero singular values of $\tilde{R}$ and $\hat{R}$, respectively, then for $i = 1, \ldots, k$*

$$0 \le \frac{\tilde{\sigma}_i - \hat{\sigma}_i}{\tilde{\sigma}_i} \le \left(\frac{\hat{\sigma}_k}{\tilde{\sigma}_i}\right) \tau \|T^{-1}\|_2.$$

*Proof.* We first note that $\max_{i=1:k} |\tilde{\sigma}_i - \hat{\sigma}_i| \le \|\tilde{R}_{[22]}\|_2 \le \tau|\tilde{r}_{kk}| \le \tau\tilde{\sigma}_1$, which estimates truncation error relative to the matrix norm. This bound actually contains better estimate because $|\tilde{r}_{kk}|$ is related to $\tilde{\sigma}_k$. From the Courant–Fisher minimax theorem we conclude $|\tilde{r}_{kk}| \le \|T^{-1}\|_2 \sigma_k(\tilde{R}_{[11]})$, where $\sigma_k(\tilde{R}_{[11]})$ is the $k$th largest singular value of $\tilde{R}_{[11]}$. The proof is completed by the Cauchy interlacing theorem, which yields $\sigma_k(\tilde{R}_{[11]}) \le \hat{\sigma}_k \le \tilde{\sigma}_k$. ☐

Consider, for example, the classical Businger–Golub pivoting [6],

$$A\Pi = Q\begin{pmatrix} R \\ 0 \end{pmatrix}, \quad |r_{ii}| \ge \sum_{k=i}^{j} r_{kj}^2, \quad 1 \le i < j \le n. \tag{2.7}$$

With careful implementation [16], the computed $\tilde{R}$ has the structure (2.7) up to small round-off. The index $k$ can be determined, for example, by looking for a gap between two consecutive diagonals of $\tilde{R}$, i.e., $|\tilde{r}_{k+1,k+1}| \le \epsilon|\tilde{r}_{kk}|$. In that case, $\|\tilde{R}_{[22]}\|_2 \le \sqrt{n-k}\epsilon|\tilde{r}_{kk}|$. Further, with $D = \operatorname{diag}(\|\tilde{R}_{[11]}(i,:)\|_2)_{i=1}^k$, the condition number $\|T^{-1}\|_2$ in Theorem 2.3 is usually smaller than $O(k)$. (Its theoretical upper bound contains $2^k$ factor, and it is attained at the Kahan matrix.) We refer to [14] for more details.

To conclude, in cases of low numerical rank, we can always use best available rank revealing QR factorization to reduce the dimension, if necessary with an additional LQ factorization, as described above.

## 3. $A$ or $A^T$?

If $A \in \mathbb{R}^{m\times n}$ with $m > n$, then the QR factorization of $A$ is an efficient preprocessor and preconditioner for the Jacobi algorithm. If $m < n$, then we start with the QR factorization of $A^T$. But what if $A$ is a square nonsingular $n \times n$ matrix? For example, let $A = DQ$, where $D$ is diagonal and $Q$ is orthogonal. In that case working with $A$ is implicit diagonalization of $A^TA = Q^TD^2Q$, while taking $A^T$ implicitly diagonalizes diagonal matrix $AA^T = D^2$. For a nonnormal $A$, a better choice between $A^TA$ and $AA^T$ has a smaller off-diagonal part and the diagonals reveal the spectrum in the sense that their distribution reveals the distribution of the spectrum as closely as possible. This desirable spectrum revealing property implies that we prefer $A$ with less equilibrated column norms. Otherwise, the effect of preconditioning is weaker, and larger angles of Jacobi rotations are more likely to appear during the process, thus causing slower convergence. In an efficient computation, the decision "$A$ or $A^T$" has to be based on at most $O(n^2)$ flops. This complexity corresponds to computing the diagonal entries of $H = A^TA$ and $M = AA^T$.

Let $s^2(H) = \sum_{i=1}^n h_{ii}^2 = \operatorname{Trace}(H \circ H)$, $s^2(M) = \operatorname{Trace}(M \circ M)$. (Here $\circ$ denotes the Hadamard matrix product.) Since $H$ and $M$ are orthogonally similar,

$\|H\|_F = \|M\|_F$, a larger value ($s^2(H)$ or $s^2(M)$) implies smaller corresponding off-norm $\Omega(H)$ or $\Omega(M)$. In fact, $s^2(\cdot)$ attains its maximum over the set of matrices orthogonally similar to $H$ only at diagonal matrices. In the standard symmetric Jacobi algorithm the value of $\frac{\Omega^2(H)}{\|H\|_F^2} = 1 - \frac{\mathrm{Trace}(H \circ H)}{\mathrm{Trace}(HH)} = 1 - \frac{s^2(H)}{\|H\|_F^2}$ is used to measure numerical convergence. Hence, $s^2(\cdot)$ is one possible choice for the decision between $A$ and $A^T$, but with respect to the standard matrix off-norm. Note, however, that in floating point computation $s^2(\cdot)$ may ignore tiny diagonal entries, and that it does not provide any information about the distributions of the diagonal entries of $H$ and $M$. The latter is crucial for the success of both the preconditioner and the Jacobi iterations. In the next section we address that issue by a novel application of well-known tools.[2]

### 3.1. Entropy of the diagonal of the adjoint orbit.

Let $H = H^T$ be positive semidefinite. From the spectral decomposition $H = U\Lambda U^T$, $h_{ii} = \sum_{j=1}^n |u_{ij}|^2 \lambda_j$, $i = 1, \ldots, n$. If we define vectors $d(H) = (\, h_{11}, \ldots, h_{nn}\,)^T$, $\lambda(H) = (\, \lambda_1, \ldots, \lambda_n\,)^T$, then $d(H) = (U \circ U)\lambda(H)$, where the matrix $S = U \circ U$ is doubly stochastic, in fact, orthostochastic. Thus, $d(H)$ is majorized by $\lambda(H)$ ($d(H) \prec \lambda(H)$), which is known as the Schur theorem; see, e.g., [3]. If we use normalization[3] by the trace,

$$\frac{d(H)}{\mathrm{Trace}(H)} = S\frac{\lambda(H)}{\mathrm{Trace}(H)}, \quad \text{and define } d'(H) = \frac{d(H)}{\mathrm{Trace}(H)}, \quad \lambda'(H) = \frac{\lambda(H)}{\mathrm{Trace}(H)},$$

then $d'(H)$ and $\lambda'(H)$ are two finite probability distributions connected by the doubly stochastic matrix $S$. Thus, $d'(H)$ has larger entropy than $\lambda'(H)$. Recall that for a probability distribution $p = (\, p_1, \ldots, p_n\,)^T$ ($p_i \geq 0$, $\sum_i p_i = 1$) the entropy of $p$ is $\eta(p) = -\frac{1}{\log n}\sum_{i=1}^n p_i \log p_i \in [0,1]$. For any doubly stochastic matrix $S$ we have $\eta(Sp) \geq \eta(p)$ with the equality if and only if $S$ is a permutation matrix. The entropy is a symmetric concave function on the compact and convex set of finite probability distributions. It is maximal, $\eta(p) = 1$, if and only if $p_i = 1/n$ for all $i$. Also, $\eta(p) = 0$ if and only if the probability distribution degenerates to $p_k = 1$, $p_i = 0$, $i \neq k$.

Definition 3.1. *The d-entropy of positive semidefinite $H$ is defined as the entropy of its diagonal normalized by the trace, $\eta_d(H) \equiv \eta(d'(H))$.*

Proposition 3.2. *The d-entropy $\eta_d$ is strictly positive on the open cone of positive definite matrices. It always attains its maximum* 1 *on the real adjoint orbit $\mathcal{O}(H) = \{W^T H W : W \text{ orthogonal}\}$. Further, it holds $\eta_d(\mathcal{O}(H)) = \{1\}$ if and only if $H$ is a scalar ($H = \mathrm{scalar} \cdot I$). If $H$ has $s$ different eigenvalues with multiplicities $n_1, \ldots, n_s$, then $\eta_d$ attains its minimal value on $\mathcal{O}(H)$ at each of $\frac{n!}{\prod_{i=1}^s n_i!}$ different diagonal matrices in $\mathcal{O}(H)$, and nowhere else.*

*Proof.* There always exists an orthogonal $W$ such that $W^T H W$ has constant diagonal. The fact that the entropy $\eta_d(\cdot)$ of the diagonal of $H$ is larger than the entropy of the vector of the eigenvalues holds for any symmetric concave function. To see that, recall the relation $d'(H) = S\lambda'(H)$, where $S$ is doubly stochastic. By the Birkhoff theorem [3, Theorem II.2.3], $S$ is from the convex hull of permutation matrices, thus $S = \sum_k \alpha_k P_k$, where the $P_k$'s are permutation matrices and the $\alpha_k$'s are nonnegative with sum one. Thus, $d'(H)$ belongs to the convex polyhedral set spanned by permutations of the vector $\lambda'(H)$. Hence, a concave function on $d'(H)$

---

[2] For the sake of brevity, we will just illustrate the main idea and leave the details for a forthcoming paper.

[3] By definition, $0/0 = 0$ and $0 \log 0 = 0$.

cannot have smaller value than its minimal value on the vectors $P_k\lambda'(H)$. Note that the number of minimal points represents the number of possible affiliations of $n$ diagonal entries with $s$ different eigenvalues. $\square$

*Example* 3.1. The following example illustrates the preceding discussion on the relation between the entropy and the spectral information along the diagonal of the matrix. Let $A$ be the upper triangular factor from the QR factorization of the $4\times 4$ Hilbert matrix, and let $H = A^TA$, $M = AA^T$. The $d(H)$, $\lambda(H)$, and $d(M)$ are

$$\begin{array}{llll} h_{11}\approx 1.4236e{+}00, & h_{22}\approx 4.6361e{-}01, & h_{33}\approx 2.4138e{-}01, & h_{44}\approx 1.5068e{-}01,\\ \lambda_1\approx 2.2506e{+}00, & \lambda_2\approx 2.8608e{-}02, & \lambda_3\approx 4.5404e{-}05, & \lambda_4\approx 9.3513e{-}09,\\ m_{11}\approx 2.2355e{+}00, & m_{22}\approx 4.3655e{-}02, & m_{33}\approx 1.3022e{-}04, & m_{44}\approx 3.5308e{-}08. \end{array}$$

If we look at only the diagonal entries of the matrix, we cannot say how close the diagonal is to the spectrum. After all, the matrix can be diagonal, that is, with minimum entropy in its orbit, and we cannot detect that. But if we have the diagonals of two orthogonally similar matrices, then we see the difference between the two diagonals. If we compute the entropies, $\eta_d(H)\approx 7.788e-001 > \eta_d(M)\approx 8.678e-002$.

*Remark* 3.1. Just looking at $d(H)$ and $d(M)$ in Example 3.1 and knowing that they are diagonals of unitarily similar matrices is enough to choose $d(M)$ as a better approximation of the spectrum since if $H$ were close to diagonal, then $\kappa_2(H)$ would be $O(1)$, while $\kappa_2(M)\geq 10^8$. In other words, orthogonal similarity can hide the high spectral condition number of diagonal matrix (so that it is not seen on the diagonal of the similar matrix), but it cannot produce it starting from a well-conditioned, almost diagonal matrix. In some sense, with respect to the problem of guessing the spectrum, the diagonal of $M$ has less uncertainty than the diagonal of $H$. Our algorithm can better utilize diagonals with a smaller entropy.

## 4. The algorithm.

We now describe the new preconditioned Jacobi SVD algorithm with rank revealing QR factorization as preconditioner. At this point we do not consider the details of the application of the one-sided Jacobi rotations to triangular matrix. Instead, we use triangular Jacobi SVD as a black box and give the details in [18]. On input to the black box we have triangular nonsingular matrix $X$, and the box computes $X_\infty = XV_x$, where $X_\infty = U_x\Sigma$, $X = U_x\Sigma V_x^T$ is the SVD of $X$, and $V_x$ is the product of the Jacobi rotations. If $V_x$ is not computed, we write $X_\infty = X\langle V_x\rangle$. We keep this notation in other situations as well. If in a relation some matrix is enclosed in $\langle\cdot\rangle$, then that matrix is not computed and no information about it is stored. For example, $A = \langle Q\rangle \begin{pmatrix} R^T & 0\end{pmatrix}^T$ means computing only $R$ in the QR factorization of $A$.

### 4.1. Computing only $\Sigma$.

We first describe the algorithm for computing only the singular values of $A$. In Algorithm 1 we use two QR factorizations with pivoting and then apply the one-sided Jacobi SVD algorithm. We do not specify which rank revealing QR factorization is used—the rule is to use the best available; see, e.g., [5], [4]. In some cases, the rows of $A$ can be sorted to get more structured backward error; see section 5.3.

*Remark* 4.1. The pivoting in the second QR factorization is optional, and $P_1 = I$ works well. If efficient QR factorization with local pivoting is available, it can be used to compute $R_1$. If $\max_{i=2:n}\|R(1:i-1,i)\|_2/|r_{ii}|\leq O(n)\varepsilon$, then the columns of $A$ are nearly orthogonal, and the second QR factorization is unnecessary, $X = R^T$.

**Algorithm 1** $\sigma = SVD(A)$.

```
$(P_r A)P = \langle Q \rangle \begin{pmatrix} R \\ 0 \end{pmatrix}$ ; $\rho = \mathrm{rank}(R)$ ; [optional deflation, section 2.2][optional $P_r$, section 5.3]
$R(1:\rho, 1:n)^T P_1 = \langle Q_1 \rangle R_1$ ; $X = R_1^T$ ; [optional pivoting $P_1$]
$X_\infty = X \langle V_x \rangle$ ; {Use section 5.4 for sharp stopping criterion.}
$\sigma_i = \|X_\infty(:, i)\|_2, \quad i = 1, \dots, \rho$ ; $\boxed{\sigma = (\sigma_1, \dots, \sigma_\rho, 0, \dots, 0)}$ .
```

**Algorithm 2** $(\sigma, V) = SVD(A)$.

```
$(P_r A)P = \langle Q \rangle \begin{pmatrix} R \\ 0 \end{pmatrix}$; $\rho = \mathrm{rank}(R)$ ; [optional deflation, section 2.2][optional $P_r$, section 5.3]
$X = R(1:\rho, 1:n)^T$ ; $X_\infty = X \langle V_x \rangle$ ;
$\sigma_i = \|X_\infty(:, i)\|_2, \quad i = 1, \dots, \rho$ ; $\boxed{\sigma = (\sigma_1, \dots, \sigma_\rho, 0, \dots, 0)}$ ;
$U_x(:, i) = \frac{1}{\sigma_i} X_\infty(:, i), \quad i = 1, \dots, \rho$ ; $\boxed{V = PU_x}$ .
```

**4.2. Computing $\boldsymbol{\Sigma}$ and $\boldsymbol{V}$.** If we need the singular values and the right singular vectors, a direct application of Jacobi rotations to $A$ or $R$ requires the accumulated product of rotations to construct the right singular vector matrix $V$. To avoid the explicit multiplication of Jacobi rotations, in this case we use Algorithm 2. The beauty of the preconditioning $R \rightsquigarrow R^T$ in Algorithm 2 is in the fact that the set of the right singular vectors is computed without the product of the Jacobi rotations, and at the same time, due to preconditioning, fewer rotations are needed to reach the numerical convergence. If $\rho \ll n$, an additional LQ factorization of $R(1:\rho, 1:n)$ is advisable. In that case, the accumulation of rotations can be avoided, as described in section 4.4.2.

**4.3. Computing $\boldsymbol{\Sigma}$ and $\boldsymbol{U}$.** If $\Sigma$ and $U$ are needed and if we apply the Jacobi SVD on $X = A$ or $X = R$, then we do not need the product of Jacobi rotations. The problem is that in the case $m \gg n$ the rotations on $A$ are too expensive, and that in both cases ($A$ or $R$) the convergence is slower than in the case $X = R^T$.

In the case $X = R^T$, we need the accumulated product of the Jacobi rotations, and the cost of the product of only one sweep of fast rotations is $2n\rho(\rho - 1) = 2n\rho^2 - 2n\rho$. To this we should also add the cost of heavier memory traffic and increased cache miss probability because two square arrays are transformed. All this is avoided by an extra QR factorization followed by transposition of the triangular factor, which is computed in $2n\rho^2 - 2\rho^3/3$ flops on BLAS 3 level.

In some cases $X = A$ is the perfect choice. For instance, if $H$ is a symmetric positive definite matrix and $P^T H P = AA^T$ its pivoted Cholesky factorization with lower triangular matrix $A$, then $A^T$ has the same properties as $R$ in (2.7). Thus $A \langle V \rangle = U\Sigma$ will be an efficient Jacobi SVD and, since $H = (PU)\Sigma^2(PU)^T$, we obtain the spectral decomposition of $H$ without computing $V$.

In Algorithm 3 we define for a matrix $M$ its property $\tau(M)$ to be *true* if $M$ is of full column rank and the Jacobi SVD algorithm applied to $M$ converges quickly. For instance, if $A$ is the Cholesky factor of positive definite matrix, computed with pivoting, then $\tau(A) = \mathit{true}$. If evaluation of $\tau(A)$ requires more than $O(mn)$ flops, or if we do not know how to judge $A$, then by definition $\tau(A) = \mathit{false}$.

**Algorithm 3** $(\sigma, U) = SVD(A)$.

```
if $\tau(A)$ then {e.g., $A$ computed by pivoted Cholesky factorization, $P^T H P = AA^T$}
    $X = A$; $X_\infty = X \langle V_x \rangle$ ;
    $\sigma_i = \|X_\infty(:,i)\|_2, \quad i = 1, \ldots, n$ ; $\boxed{\sigma = (\sigma_1, \ldots, \sigma_n)}$ ;
    $\boxed{U(:,i) = \frac{1}{\sigma_i} X_\infty(:,i), \quad i = 1, \ldots, n}$ ;
else
    $(P_r A)P = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$; $\rho = \mathrm{rank}(R)$ ; [optional deflation, section 2.2][optional $P_r$, section 5.3]
    if $\tau(R)$ then {e.g., $\max_{i=2:n} \|R(1:i-1,i)\|_2/|r_{ii}|$ small}
        $X = R$ ; $X_\infty = X \langle V_x \rangle$ ;
        $\sigma_i = \|X_\infty(:,i)\|_2, \quad i = 1, \ldots, \rho$ ; $\boxed{\sigma = (\sigma_1, \ldots, \sigma_\rho, 0, \ldots, 0)}$ ;
        $U_x(:,i) = \frac{1}{\sigma_i} X_\infty(:,i), \quad i = 1, \ldots, \rho$; $\boxed{U = P_r^T Q \begin{pmatrix} U_x \\ 0_{(m-\rho)\times\rho} \end{pmatrix}}$ ;
    else
        $R(1:\rho, 1:n)^T P_1 = \langle Q_1 \rangle R_1$ ; [optional pivoting $P_1$]
        $X = R_1^T$ ; $X_\infty = X \langle V_x \rangle$ ;
        $\sigma_i = \|X_\infty(:,i)\|_2, \quad i = 1, \ldots, \rho$ ; $\boxed{\sigma = (\sigma_1, \ldots, \sigma_\rho, 0, \ldots, 0)}$ ;
        $U_x(:,i) = \frac{1}{\sigma_i} X_\infty(:,i), \quad i = 1, \ldots, \rho$; $\boxed{U = P_r^T Q \begin{pmatrix} P_1 U_x \\ 0_{(m-\rho)\times\rho} \end{pmatrix}}$ ;
    end if
end if
```

### 4.4. Computation of $U$, $\Sigma$, and $V$.

In this section we describe an efficient implementation of the preconditioned Jacobi SVD algorithm for computing the full SVD. The classical implementation of the Jacobi SVD algorithm transforms two matrices, one approaching the matrix of the left singular vectors scaled by the corresponding singular values ($U\Sigma$), and the second being the accumulated product of the Jacobi rotations ($V$). We extend an idea from [14] and compute the product of Jacobi rotations a posteriori from a well-conditioned matrix equation. In this way, the expensive iterative part has fewer flops and needs less cache space. The rotations are explicitly accumulated only if none of four candidate matrix equations can guarantee an accurate solution.

#### 4.4.1. Classical computation of $V$ by accumulation.

Let the Jacobi iterations stop at index $\overline{k}$ and let $\tilde{X}_\infty = \tilde{X}^{(\overline{k})}$. Let $\tilde{V}_x$ be the computed accumulated product of Jacobi rotations used to compute $\tilde{X}_\infty$. Rowwise backward stability implies that $\tilde{X}_\infty = (X + \delta X)\hat{V}_x$, where $\hat{V}_x$ is orthogonal, and $\|\delta X(i,:)\|_2 \leq \epsilon_J \|X(i,:)\|_2$, $\epsilon_J \leq O(n\varepsilon)$; see [13]. The matrix $\tilde{V}_x$ can be written as $\tilde{V}_x = (I + E_0)\hat{V}_x$, where $\|E_0\|_2$ is small. In fact, $\max_i \|E_0(i,:)\|_2 \leq \varepsilon_J$. Note that the matrix $\hat{V}_x$ is a purely theoretical entity—it exists only in the proof of the backward stability. If we want to recover $\hat{V}_x$, the best we can do is to compute

$$X^{-1}\tilde{X}_\infty = (I + E_1)\hat{V}_x, \quad E_1 = X^{-1}\delta X, \tag{4.1}$$

since we do not have $\delta X$. Thus, we can come $\|E_1\|_2$ close to $\hat{V}_x$. To estimate $E_1$, we write $X = DY$, where $D$ is diagonal scaling, $D_{ii} = \|X(i,:)\|_2$, and $Y$ has unit rows in the Euclidean norm. We obtain for $\|\cdot\| \in \{\|\cdot\|_2, \|\cdot\|_F\}$

$$\|E_1\| \le \|Y^{-1}\|_2 \|D^{-1}\delta X\| \le \|Y^{-1}\|_2 \sqrt{n}\epsilon_J \le \|Y^{-1}\|_2 O(n^{3/2})\boldsymbol{\varepsilon}. \tag{4.2}$$

Finally, the matrix $\tilde{X}_\infty$ is written as $\tilde{U}_x\tilde{\Sigma}$. The diagonal entries of $\tilde{\Sigma}$ are computed as $\tilde{\Sigma}_{ii} = \tilde{\sigma}_i = \mathit{computed}(\|\tilde{X}_\infty(:,i)\|_2) = \|\tilde{X}_\infty(:,i)\|_2(1+\nu_i)$, $|\nu_i| \le O(n\boldsymbol{\varepsilon})$, and then $\tilde{U}_x(:,i)$ is computed by dividing $\tilde{X}_\infty(:,i)$ by $\tilde{\sigma}_i$. Thus,

$$\tilde{U}_x\tilde{\Sigma} = \tilde{X}_\infty + \delta\tilde{X}_\infty, \quad |\delta\tilde{X}_\infty| \le \boldsymbol{\varepsilon}|\tilde{X}_\infty|. \tag{4.3}$$

If $\tilde{\sigma}_i$ is computed using a double accumulated dot product, then $|\nu_i| \le O(\boldsymbol{\varepsilon})$ and the columns of $\tilde{U}_x$ are of unit norm up to $O(\boldsymbol{\varepsilon})$. The following proposition explains how well the computed SVD resembles the matrix $X$.

PROPOSITION 4.1. *The matrices* $\tilde{U}_x$, $\tilde{\Sigma}$, $\tilde{V}_x$, $\hat{V}_x$ *satisfy residual relations*

$$\text{(i)}\quad \tilde{U}_x\tilde{\Sigma}\hat{V}_x^T = X + F = X(I + X^{-1}F); \quad \text{(ii)}\quad \tilde{U}_x\tilde{\Sigma}\tilde{V}_x^T = (X+F)(I+E_0^T),$$

*where for all* $i$, $\|F(i,:)\|_2 \le \overline{\varepsilon_J}\|X(i,:)\|_2$, $\overline{\varepsilon_J} = \varepsilon_J + \boldsymbol{\varepsilon}(1+\varepsilon_J)$. *Further,* $\|E_0\|_2 \le \sqrt{n}\varepsilon_J \le O(n^{3/2}\boldsymbol{\varepsilon})$, *and* $\|X^{-1}F\|_2 \le \|Y^{-1}\|_2\sqrt{n}\overline{\varepsilon_J}$.

*Proof.* From the relations $\tilde{X}_\infty = (X+\delta X)\hat{V}_x$ and (4.3) we obtain $\tilde{U}_x\tilde{\Sigma}\hat{V}_x^T = X+F$, $F = \delta X + \delta\tilde{X}_\infty\hat{V}_x^T$, and for (ii) we use $\tilde{V}_x = (I+E_0)\hat{V}_x$. ☐

**4.4.2. Computation of $V$ from matrix equation.** Suppose that, instead of $\tilde{V}_x$, we decide to use some other approximation of $\hat{V}_x$. The matrix $X^{-1}\tilde{X}_\infty$ is a good candidate (it gets $\|E_1\|_2$ close to $\hat{V}_x$), but we cannot have the exact value of $X^{-1}\tilde{X}_\infty$. We can solve the matrix equation and take $\breve{V}_x = \mathit{computed}(X^{-1}\tilde{X}_\infty)$. Since $X$ is triangular, the residual bound for $\breve{V}_x$ is

$$E_2 = X\breve{V}_x - \tilde{X}_\infty, \quad |E_2| \le \epsilon_T|X||\breve{V}_x|, \quad \epsilon_T \le \frac{n\boldsymbol{\varepsilon}}{1-n\boldsymbol{\varepsilon}}. \tag{4.4}$$

From (4.1) and (4.4) we conclude that

$$\breve{V}_x = (I+E_3)\hat{V}_x = \hat{V}_x(I + \hat{V}_x^T E_3 \hat{V}_x), \quad E_3 = E_1 + X^{-1}E_2\hat{V}_x^T, \tag{4.5}$$

where only the symmetric part $Sym(E_3) = 0.5(E_3 + E_3^T)$ contributes to the first order departure from orthogonality of $\breve{V}_x$, $\|\breve{V}_x^T\breve{V}_x - I\|_2 \le 2\|Sym(E_3)\|_2 + \|E_3\|_2^2$.

The following proposition shows that we have also computed a rank revealing decomposition of $X$ (in the sense of [10]).

PROPOSITION 4.2. *The matrices* $\tilde{U}_x$, $\tilde{\Sigma}$, $\breve{V}_x$ *satisfy the residual relations*

$$\tilde{U}_x\tilde{\Sigma}\breve{V}_x^T = (X+F)(I+E_3^T), \quad E_3 = E_1 + X^{-1}E_2\hat{V}_x^T; \tag{4.6}$$

$$\tilde{U}_x\tilde{\Sigma}\breve{V}_x^{-1} = X + F_1, \quad F_1 = E_2\breve{V}_x^{-1} + \delta\tilde{X}_\infty\breve{V}_x^{-1}, \tag{4.7}$$

*where* $F$ *is as in Proposition* 4.1, $\|E_3\|_2 \le \|Y^{-1}\|_2(\sqrt{n}\varepsilon_J + n\epsilon_T)$, *and it holds for all* $i$ *that* $\|F_1(i,:)\|_2 \le (\varepsilon_T\||\breve{V}_x|\|_2 + \boldsymbol{\varepsilon}(1+\varepsilon_J))\|\breve{V}_x^{-1}\|_2\|X(i,:)\|_2$.

This analysis shows that the quality of the computed right singular vector matrix $\breve{V}_x$ depends on the condition number $\|Y^{-1}\|_2$, where $X = DY$. Hence, the rows of the triangular matrix $X$ must be well-conditioned in the scaled sense. If $X$ is computed from the initial $A$ using the QR factorization with column pivoting, $AP = Q\begin{pmatrix} R^T & 0 \end{pmatrix}^T$, then $X = R$ can be written as $X = DY$ with well-conditioned $Y$.

Thus, we expect that $\check{V}_x$ can be computed accurately, but immediately notice a drawback. The Jacobi rotations implicitly transform the matrix $P^T(A^TA)P$, which means that we do not have the preconditioning effect—for that the input matrix to Jacobi procedure should be $X^T = Y^TD$.

We conclude that the initial matrix should be of the form $X = DY = ZC$, where $D$, $C$ are diagonal and both $Y$ and $Z$ are well-conditioned. Well-conditioned $Z$ implies fast convergence, while well-conditioned $Y$ ensures stable a posteriori computation of the right singular vectors. Therefore, we define $X$ in the following way:

(i) $AP = Q\begin{pmatrix} R^T & 0 \end{pmatrix}^T$;
(ii) $R^TP_1 = Q_1R_1$;
(iii) $X = R_1^T$.

The matrix $R$ can be written as $R = D_rR_r$ with well-conditioned $R_r$, and if we write $R_1 = (R_1)_c(D_1)_c$, then $\kappa((R_1)_c) = \kappa(R_r)$; thus $X = DY$ with $D = (D_1)_c$, $Y = (R_1)_c^T$. Further, $R_1 = (D_1)_r(R_1)_r$ with the expected value of $\kappa((R_1)_r)$ smaller than $\kappa((R_1)_c)$, and thus $X = ZD_c$ with well-conditioned $Z$. In fact, $Z^TZ$ is very strongly diagonally dominant. We have strong numerical evidence that the pivoting in the second QR factorization is not worth the overhead. If we have an efficient QR factorization with local pivoting, such overhead is negligible. Note that $X = R_1$ also has the required properties. Without column pivoting in the second QR factorization ($P_1 = I$) we cannot give any theoretical bound on the condition number of $Y$, and condition estimators must be used. Putting all of this together, we obtain Algorithm 4.

Since the key matrices in the algorithm are all triangular, condition estimators can be used to control the program flow. We can decide which matrix is the best input to the one-sided Jacobi algorithm, or which matrix equation to solve. For instance, in the case $\rho = n$ and small $\kappa_1$, the SVD $R_1^T = U_x\Sigma V_x^T$ implies $V_x = R_1^{-T}(U_x\Sigma)$, but we also note that $R(Q_1V_x) = (U_x\Sigma)$. It can be shown (as in section 4.4.2) that computing $W = Q_1V_x$ very efficiently as $R^{-1}X_\infty$ is numerically as accurate as first computing $V_x = R_1^{-T}X_\infty$ and then multiplying $Q_1V_x$. (Similar situations occur in the case of well-conditioned $Y$ and $X = L_2$, where $Q_2^TV_x$ is computed directly as $R_1^{-1}X_\infty$.) Since in each major step we have estimates of relevant condition number (of scaled matrices), the algorithm can be authorized (an input option) to drop some small singular values if the condition number test shows that they are highly sensitive.

The last line of defense in Algorithm 4 computes with explicit accumulation of Jacobi rotations. So far, we know of no example in which accumulation of Jacobi rotation is needed, because the previous three preconditioning steps failed to produce $X$, which is structured as $X = DY$ with moderate $\|Y^{-1}\|_2$. In fact, we never had the case that required $X = L_2^T$. The worst-case example, which probably already has crossed the reader's mind, is Kahan's matrix [26].

*Example* 4.1. It is instructive to see how our algorithm deals with the upper triangular Kahan's matrix $K = K(m,c)$ with $K_{ii} = s^{i-1}$ and $K_{ij} = -c\cdot s^{i-1}$ for $i < j$, where $s^2 + c^2 = 1$. Using MATLAB, we generate $K(100, 0.9998)$. It is estimated that $\kappa_1 \approx \|R_r^{-1}\|$ is bigger than $10^{16}$. Now, the trick here is that our entropy test will transpose the matrix automatically and take $A = K^T$ instead of $A = K$. In that case the estimated $\kappa_1$ is around one. Suppose now that the transposing mechanism is switched off, or that, e.g., $A = K(1:m, 1:n)$, $n < m$, so that no transposition is allowed. Let $A$ be equal to the first 90 columns of $K$. Again, $\kappa_1 > 10^{16}$, but $\kappa_Y \approx 1$.

## 5. Assessing the accuracy of the computed SVD.

In this section we analyze numerical properties of the new algorithm. To simplify the notation, we drop the permutation matrices, thus assuming that $A$ is replaced with the permuted matrix

**Algorithm 4** $(U, \sigma, V) = \mathrm{SVD}(A)$.

```
$(P_rA)P = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$ ; $\rho = \mathrm{rank}(R)$ ; [optional deflation, section 2.2][optional $P_r$, section 5.3]
if $\max_{i=2:n} \|R(1:i-1,i)\|/|r_{ii}|$ small then {columns of $A$ almost orthogonal}
  $X = R$ ; $\boxed{\kappa_0 = \mathit{estimate}(\|A_c^\dagger\|_2)}$ ; {At this point, $\kappa_0 \ll n$. $A^TA$ is $\gamma$-s.d.d.}
  $X_\infty = X\,\langle V_x\rangle$ ; $V_x = R^{-1}X_\infty$ ; $\boxed{\sigma_i = \|X_\infty(:,i)\|_2, \quad i = 1,\ldots,n}$ ; $\boxed{V = PV_x}$ ;
  $U_x(:,i) = \frac{1}{\sigma_i}X_\infty(:,i), \quad i = 1,\ldots,n$ ; $\boxed{U = P_r^TQ\begin{pmatrix} U_x & 0 \\ 0 & I_{m-n} \end{pmatrix}}$ ;
else
  $\boxed{\kappa_0 = \mathit{estimate}(\|A_c^\dagger\|_2)}$ ; $\kappa_1 = \mathit{estimate}(\|R_r^\dagger\|_2)$ ;
  if $\kappa_1$ small then {e.g., $\kappa_1$ small $\Longleftrightarrow \kappa_1 < \sqrt{n}$, or, e.g., $\kappa_1 < n$}
    $R(1:\rho, 1:n)^T = Q_1\begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ {second preconditioning}; $X = R_1^T$;
  else
    $R(1:\rho, 1:n)^TP_1 = Q_1\begin{pmatrix} R_1 \\ 0 \end{pmatrix}$ {second preconditioning};
    $R_1 = L_2\,\langle Q_2\rangle$ {third preconditioning: LQ factorization}; $X = L_2$;
    $\kappa_Y = \mathit{estimate}(\|Y^{-1}\|_2)$ ;   if $\kappa_Y \geq n$ then $\kappa_Z = \mathit{estimate}(\|Z^{-1}\|_2)$ ; end if
  end if
  if $Y$ well conditioned then
    $X_\infty = X\,\langle V_x\rangle$; $\boxed{\sigma_i = \|X_\infty(:,i)\|_2}$ ; $U_x(:,i) = \frac{1}{\sigma_i}X_\infty(:,i), \quad i = 1,\ldots,\rho$;
    if $\rho = n$ and $\kappa_1$ small then
      $W = R^{-1}X_\infty$; {here $W \equiv Q_1V_x$}; $\boxed{V = PW}$ ; $\boxed{U = P_r^TQ\begin{pmatrix} U_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$ ;
    else if $\kappa_1$ small then {$R$ rectangular, $\rho < n$}
      $V_x = R_1^{-T}X_\infty$; $\boxed{V = PQ_1\begin{pmatrix} V_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$ ; $\boxed{U = P_r^TQ\begin{pmatrix} U_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$ ;
    else {here $X = L_2$ and $W \equiv Q_2^TV_x$}
      $W = R_1^{-1}X_\infty$; $\boxed{V = PQ_1\begin{pmatrix} U_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$ ; $\boxed{U = P_r^TQ\begin{pmatrix} P_1W & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$ ;
    end if
  else if $\kappa_Z < n$ then
    $X = L_2^T$; $X_\infty = X\,\langle V_x\rangle$; $\boxed{\sigma_i = \|X_\infty(:,i)\|_2}$ ; $U_x(:,i) = \frac{1}{\sigma_i}X_\infty(:,i)$, $i = 1,\ldots,\rho$;
    $V_x = L_2^{-T}X_\infty$ ; $\boxed{V = PQ_1\begin{pmatrix} V_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$ ; $\boxed{U = P_r^TQ\begin{pmatrix} P_1Q_2^TU_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$ ;
  else {last line of defense: use $X = L_2$ and accumulate Jacobi rotations}
    $X_\infty = XV_x$ ; $\boxed{\sigma_i = \|X_\infty(:,i)\|_2}$ ; $U_x(:,i) = \frac{1}{\sigma_i}X_\infty(:,i)$, $i = 1,\ldots,\rho$ ;
    $\boxed{V = PQ_1\begin{pmatrix} U_x & 0 \\ 0 & I_{n-\rho} \end{pmatrix}}$ ; $\boxed{U = P_r^TQ\begin{pmatrix} P_1Q_2^TV_x & 0 \\ 0 & I_{m-\rho} \end{pmatrix}}$ ;
  end if
end if
```

$P_rAP$. The computed matrices are marked by *tildes*, and by *hats* we denote matrices whose existence is obtained during backward error analysis (they are usually close to the corresponding matrices marked with tildes). We note that detailed analysis is given without the details of the triangular SVD computation $X_\infty = XV_x$. We only need the fact that the computed $\tilde{X}_\infty$, $\tilde{V}_x$ satisfy $\tilde{X}_\infty = (X + \delta X)\hat{V}_x$, where for all $i$ $\|\delta X(i,:)\|_2 \leq \varepsilon_J \|X(i,:)\|_2$, $\varepsilon_J \leq O(n\boldsymbol{\varepsilon})$, and $\hat{V}_x$ is exactly orthogonal, close to $\tilde{V}_x$. This is independent of the pivot strategy. For the sake of brevity we will not analyze all variants of algorithms given in section 4.

### 5.1. Backward error analysis.

The following proposition is central to the analysis of Algorithms 1 and 2. It gives backward stability with a rather strong columnwise estimate of the relative backward error.

PROPOSITION 5.1. *Let the SVD of the real $m \times n$ matrix $A$ be computed by reducing $A$ to triangular form, $A = Q\begin{pmatrix} R^T & 0\end{pmatrix}^T$, and then applying the Jacobi SVD algorithm to $X = R^T$. If only the singular values or singular values and the right singular vectors are needed (Algorithms 1 or 2), then the backward stability of the computation can be described as follows:*

(i) *Let $X \approx \tilde{U}_x \tilde{\Sigma} \langle \tilde{V}_x^T \rangle$ be the computed SVD of the computed matrix $X$. Then there exist a columnwise small perturbation $\Delta A$ and orthogonal matrices $\hat{Q}$, $\hat{V}_x$ such that*

$$A + \Delta A = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\tilde{U}_x^T, \quad \textit{where} \tag{5.1}$$

$$\|\Delta A(:,i)\|_2 \leq \tilde{\eta}\|A(:,i)\|_2, \;\; i = 1, \ldots, n, \;\; \tilde{\eta} = \varepsilon_{qr} + \overline{\varepsilon_J} + \varepsilon_{qr}\overline{\varepsilon_J}. \tag{5.2}$$

(ii) *Furthermore, let $\varepsilon_u \equiv \|\tilde{U}_x^T\tilde{U}_x - I\|_F < 1/2$. There exist a backward perturbation $\mathcal{E}$ and orthogonal matrix $\hat{U}$ such that $\|\tilde{U}_x - \hat{U}\|_F \leq \sqrt{2}\varepsilon_u$ and the SVD of $A + \mathcal{E}$ is*

$$A + \mathcal{E} = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\hat{U} \equiv \hat{U}_a\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\hat{V}_a^T, \quad \textit{where} \tag{5.3}$$

$$\|\mathcal{E}(:,i)\|_2 \leq \hat{\eta}\|A(:,i)\|_2, \;\; \hat{\eta} = \tilde{\eta} + \sqrt{2n}\varepsilon_u + O(\boldsymbol{\varepsilon}^2) \quad \textit{for all } i. \tag{5.4}$$

*Proof.* Let $\tilde{Q}$ and $\tilde{R}$ be the computed numerically orthogonal and the triangular factor of $A$, respectively. Then there exist an orthogonal matrix $\hat{Q}$ and backward perturbation $\delta A$ such that $A + \delta A = \hat{Q}\begin{pmatrix} \tilde{R}^T & 0 \end{pmatrix}^T$, where for all column indices $\|\delta A(:,i)\|_2 \leq \varepsilon_{qr}\|A(:,i)\|_2$. Let the one-sided Jacobi SVD be applied to $X = \tilde{R}^T$. By Proposition 4.1, $X + F = \tilde{U}_x\tilde{\Sigma}\hat{V}_x^T$, $\|F(i,:)\|_2 \leq \overline{\varepsilon_J}\|X(i,:)\|_2$, and therefore

$$A + \delta A + \hat{Q}\begin{pmatrix} F^T \\ 0 \end{pmatrix} = \hat{Q}\begin{pmatrix} \hat{V}_x & 0 \\ 0 & I \end{pmatrix}\begin{pmatrix} \tilde{\Sigma} \\ 0 \end{pmatrix}\tilde{U}_x^T, \;\; \Delta A = \delta A + \hat{Q}\begin{pmatrix} F^T \\ 0 \end{pmatrix}, \tag{5.5}$$

where $\|\Delta A(:,i)\|_2 \leq \varepsilon_{qr}\|A(:,i)\|_2 + \overline{\varepsilon_J}\|\tilde{R}(:,i)\|_2$, $\|\tilde{R}(:,i)\|_2 \leq (1 + \varepsilon_{qr})\|A(:,i)\|_2$, and (5.1), (5.2) follow. Note that the right-hand side of relation (5.1) is not an SVD. To obtain a relation with the SVD of a matrix in the vicinity of $A$, we need to replace $\tilde{U}_x$ with a nearby orthogonal matrix. However, since the backward error $\Delta A$ is columnwise small, we need to do this carefully and preserve this fine structure of the backward error. Since $\tilde{U}_x$ is on the right-hand side, correcting its departure from orthogonality implies certain linear combinations of the columns of $A + \Delta A$. If $A$ has very large and very small columns, then such linear combinations may introduce large perturbations into the small ones. This is why we cannot use the orthogonal polar factor of $\tilde{U}_x$ as the closest orthogonal matrix. We proceed with the following thought experiment.

Let $\Pi$ be a permutation matrix such that the columns of $A\Pi$ have decreasing Euclidean lengths.[4] Let $\Pi^T\tilde{U}_x = (I + G_0^T)\hat{U}_x$ be the RQ factorization of $\Pi^T\tilde{U}_x$, with lower triangular $G_0$ and orthogonal $\hat{U}_x$. Since $\tilde{U}_x$ is numerically orthogonal, we can nicely estimate $G_0$. First, $I + G_0$ is regular.

From the Cholesky factorization $(I + G_0)(I + G_0)^T = I + \hat{U}_x(\tilde{U}_x^T\tilde{U}_x - I)\hat{U}_x^T$, we conclude, using [17], that $\|G_0\|_F \leq \sqrt{2}\varepsilon_u$. Let $I + G = (I + G_0)^{-1}$. Obviously, $G$ is lower triangular. Since $G = -G_0 + G_0^2(I + G_0)^{-1}$, it holds that $\|G\|_1 \leq \|G_0\|_1 + \|G_0\|_1^2/(1 - \|G_0\|_1)$. From (5.5) we obtain the SVD

$$(A + \Delta A)(I + \Pi G\Pi^T) = \hat{Q}\begin{pmatrix}\hat{V}_x & 0\\ 0 & I\end{pmatrix}\begin{pmatrix}\tilde{\Sigma}\\ 0\end{pmatrix}(\Pi\hat{U}_x)^T. \tag{5.6}$$

Note that small $\|\Pi G\Pi^T\|_1 = \|G\|_1 \approx \|G_0\|_1$ does not automatically mean columnwise small backward perturbation in $A$. Let us estimate the columns of $A\Pi G\Pi^T$. Note that in the multiplication $A\Pi G$ each column of $A$ gets a contribution only from columns that are smaller in norm, i.e., $A\Pi G(:,i) = \sum_{k=i}^{n} g_{ki}(A\Pi)(:,k)$, and thus $\|A\Pi G(:,i)\|_2 \leq \|G\|_1\|(A\Pi)(:,i)\|_2$. Since $\Pi^T$ redistributes the columns back to the original order, we have $\|(A\Pi G\Pi^T)(:,i)\|_2 \leq \|G\|_1\|A(:,i)\|_2$.

Similarly, $\|(\Delta A\Pi G\Pi^T)(:,i)\|_2 \leq \tilde{\eta}\|G\|_1\|A(:,i)\|_2$. Note that from the relation $\tilde{U}_x = (I + \Pi G_0^T\Pi^T)(\Pi\hat{U}_x)$ we easily find that the matrix $\hat{U} = \Pi\hat{U}_x$ satisfies $\|\hat{U} - \tilde{U}_x\|_F \leq \|G_0\|_F$. Finally, note that (5.6) defines $\mathcal{E}$ in relation (5.3). ☐

Consider now the computation of the full SVD with a single preconditioning step.

PROPOSITION 5.2. *Let* $A \approx \tilde{Q}\binom{\tilde{R}}{0}$ *be the computed QR factorization of* $A$. *Let the computed SVD of* $X = \tilde{R}^T$ *be* $X \approx \tilde{U}_x\tilde{\Sigma}\overline{V}_x$, *where the following hold.*

(a) $\overline{V}_x = \tilde{V}_x$ *if* $\overline{V}_x$ *is computed as the accumulated product of Jacobi rotations (Proposition* 4.1*). In that case* $\|\overline{V}_x - \hat{V}_x\|_F \leq \sqrt{n}\varepsilon_J$.

(b) $\overline{V}_x = \breve{V}_x$ *if* $\overline{V}_x$ *is computed from the triangular matrix equation (Proposition* 4.2*). In that case* $\|\overline{V}_x - \hat{V}_x\|_F \leq \|Y^{-1}\|_2(\sqrt{n}\varepsilon_J + n\varepsilon_T)$, *where* $Y = \mathrm{diag}(1/\|X(i,:)\|_2)X$.

*Let* $\tilde{V}_a = \tilde{U}_x$, $\hat{V}_a = \hat{U}$, *where* $\hat{U}$ *is as in Proposition* 5.1 *and let*

$$\hat{U}_a = \hat{Q}\begin{pmatrix}\hat{V}_x & 0\\ 0 & I\end{pmatrix}, \quad \tilde{U}_a = computed\left(\tilde{Q}\begin{pmatrix}\overline{V}_x & 0\\ 0 & I\end{pmatrix}\right).$$

*Then* $\|\tilde{U}_a - \hat{U}_a\|_2 \leq \sqrt{m}\varepsilon_{qr} + \|\overline{V}_x - \hat{V}_x\|_2 + O(\varepsilon^2)$, $\|\tilde{V}_a - \hat{V}_a\|_F \leq \sqrt{2}\varepsilon_u$, *and the residual (that is, the backward error)*

$$\Delta' A = \tilde{U}_a\begin{pmatrix}\tilde{\Sigma}\\ 0\end{pmatrix}\tilde{V}_a^T - A = (I + (\tilde{U}_a - \hat{U}_a)\hat{U}_a^T)(A + \Delta A) - A \tag{5.7}$$

*satisfies* $\|\Delta' A(:,i)\|_2 \leq \tilde{\eta}'\|A(:,i)\|_2$, $\quad \tilde{\eta}' = \tilde{\eta} + \|\tilde{U}_a - \hat{U}_a\|_2 + \tilde{\eta}\|\tilde{U}_a - \hat{U}_a\|_2$.

*Proof.* In addition to the proof of Proposition 5.1, we need an estimate for $\tilde{U}_a - \hat{U}_a$. Note that $\tilde{U}_a$ is computed using Householder vectors computed in the QR factorization, and then replace $\overline{V}_x$ with $\hat{V}_x + (\overline{V}_x - \hat{V}_x)$. ☐

---

[4] One should keep in mind that $\Pi$ is an object in our thought experiment, unrelated to actual pivoting in the algorithm.

**5.2. Backward errors for two preconditionings.** In the case $X = R^T$ all transformations are applied to $A$ from the same side, $\binom{\Sigma}{0}U_x^T = (V_x^T \oplus I)Q^T A$. This fact is the key for columnwise small backward error. In the case of two QR factorizations in the preconditioning phase, where $X = R_1^T$, columnwise small backward error in $A$ is not obvious, because in that case we compute ULV decomposition by transforming $A$ from both sides.

THEOREM 5.3. *For the computed matrix* $\tilde{X}_\infty \approx \tilde{U}_x\tilde{\Sigma}$, *there exist backward perturbation* $\Delta A$, *permutation* $\tilde{P}_1$, *and orthogonal* $\hat{Q}$, $\hat{Q}_1$, $\hat{V}_x$, $S$ *such that*

$$\begin{pmatrix} \tilde{U}_x\tilde{\Sigma} \\ 0 \end{pmatrix} \approx \begin{pmatrix} \tilde{X}_\infty \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{P}_1^T & 0 \\ 0 & I_{m-n} \end{pmatrix} \hat{Q}^T (A + \Delta A) S \hat{Q}_1 \hat{V}_x^T. \tag{5.8}$$

*For each* $i$, $\Delta A(:,i)$ *is a small relative perturbation of* $A(:,i)$, *and* $S$ *is close to identity.*

*Proof.* The second factorization $R^T P_1 = Q_1 R_1$ is computed as $(\tilde{R}^T + \delta\tilde{R}^T)\tilde{P}_1 = \hat{Q}_1\tilde{R}_1$, where $\|\delta\tilde{R}(i,:)\|_2 \le \varepsilon_{qr}\|\tilde{R}(i,:)\|_2$. Jacobi rotations are applied to $X = \tilde{R}_1^T$, which yields $\tilde{X}_\infty = (X + \delta X)\hat{V}_x$, $\|\delta X(i,:)\|_2 \le \varepsilon_J\|X(i,:)\|_2$. This means that $\tilde{R}_1$ is changed backward to $\tilde{R}_1 + \delta\tilde{R}_1$ with columnwise bound $\|\delta\tilde{R}_1(:,i)\|_2 \le \varepsilon_J\|\tilde{R}_1(:,i)\|_2$. To push $\delta\tilde{R}_1$ further backward we have to change $\tilde{R}$. It is easy to check that $\Delta\tilde{R} = \delta\tilde{R} + \tilde{P}_1\delta\tilde{R}_1^T\hat{Q}_1^T$ is a rowwise small perturbation of $\tilde{R}$ with the property

$$(\tilde{R}^T + \Delta\tilde{R}^T)\tilde{P}_1 = \hat{Q}_1(\tilde{R}_1 + \delta\tilde{R}_1). \tag{5.9}$$

Write $\tilde{R} + \Delta\tilde{R} = \tilde{R}(I + E)$ with $E = \tilde{R}^{-1}\Delta\tilde{R}$, and let $\tilde{R} = D_r\tilde{R}_r$ with $D_r = \mathrm{diag}(\|\tilde{R}(i,:)\|_2)_{i=1}^n$. It is easily shown that $\|E\|_F \le \sqrt{n}(\varepsilon_{qr} + \varepsilon_J(1 + \varepsilon_{qr}))\|\tilde{R}_r^{-1}\|_2$. Note that this bound depends on $\|\tilde{R}_r^{-1}\|_2$, which in our case is at most $O(n)$.

Let $\Delta A_0 = \delta A + \hat{Q}\left( (\Delta\tilde{R})^T \quad 0 \right)^T$. Then we almost have the explicit backward relationship (5.8) with columnwise bound. The backward perturbed matrix is

$$A + \Delta_0 A = (A + \delta A)(I + E) \quad (= (I + \delta A A^\dagger)A(I + E) \text{ if } \mathrm{rank}(A) = n). \tag{5.10}$$

Note that $I + E$ represents multiplicative backward perturbation, which immediately and cleanly exposes its corresponding forward error. However, additive backward perturbation might be more desirable and interpretable. Therefore, we are going to transform the multiplicative part into an additive one. If the columns of $A + \delta A$ are not ordered from large to small in the Euclidean norm, then we order them using permutation $\Pi$ and write $(A + \delta A)(I + E) = (A + \delta A)\Pi(I + \Pi^T E \Pi)\Pi^T$.

If $I + \Pi^T E \Pi = L S_0$ is the LQ factorization, then we can write $L = I + F$ with lower triangular $F$ and $\|F\|_F \le O(1)\|E\|_F$. The orthogonal matrix $S_0$ is close to identity. Then we have

$$(A + \delta A)(I + E) = (A + \delta A)\Pi(I + F)S_0\Pi^T = ((A + \delta A)\Pi + (A + \delta A)\Pi F)S_0\Pi^T,$$

where $\|((A + \delta A)\Pi F)(:,i)\|_2 \le \|F\|_1\|((A + \delta A)\Pi)(:,i)\|_2$. If we permute the columns of $A + \delta A$ back to the original order, we obtain

$$A + \Delta A_0 = (A + \delta A)(I + E) = (A + \delta A + \delta_1 A)\Pi S_0 \Pi^T, \tag{5.11}$$

where $\|\delta_1 A(:,i)\|_2 \le (1 + \varepsilon_{qr}(A))\|F\|_1\|A(:,i)\|_2$, $i = 1, \ldots, n$. Using this in (5.8), we conclude that $\tilde{U}_x\tilde{\Sigma}$ is computed by orthogonal transformations on $A + \delta A + \delta_1 A$, where the perturbation $\Delta A = \delta A + \delta_1 A$ is columnwise small, and $S = \Pi S_0 \Pi^T$. ☐

The practical value of this is that no matter how the columns of $A$ are scaled, the algorithm computes the SVD of $A$ with columnwise small backward relative error.

**5.3. Complete pivoting and two-sided scaling.** In [14], we recommended that the rows of $A$ be sorted before the first QR factorization with column pivoting, thus having the effect of complete pivoting suggested by Powell and Reid [30]. The reason was more structured backward error, as shown in [9]. It is important that the whole algorithm preserves this structured perturbation.

THEOREM 5.4. *Let $A = D_1 C D_2$, where $D_1$, $D_2$ are diagonal matrices, be prepivoted so that the computed QR factorization satisfies*

$$D_1(C + \delta C)D_2 = \hat{Q}\begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix}, \quad \hat{Q}^T\hat{Q} = I_m. \tag{5.12}$$

*Let $q = \max_{i\geq j} |(D_2)_{ii}/(D_2)_{jj}|$. There exists an orthogonal matrix $S$, close to identity, such that the backward perturbation in Theorem* 5.3 *can be written as*

$$\begin{pmatrix} \tilde{U}_x\tilde{\Sigma} \\ 0 \end{pmatrix} \approx \begin{pmatrix} \tilde{X}_\infty \\ 0 \end{pmatrix} = \begin{pmatrix} \tilde{P}_1^T & 0 \\ 0 & I_{m-n} \end{pmatrix} \hat{Q}^T D_1(C + \Delta C)D_2 S\hat{Q}_1\hat{V}_x^T. \tag{5.13}$$

*It holds that $\|\Delta C\|_F \leq \|\delta C\|_F + q\sqrt{8}\|E\|_F(\|C\|_2 + \|\delta C\|_2) + O(\|E\|_F^2)$.*

*Proof.* We go back to relation (5.10) and rewrite it as

$$A + \Delta A = (A + \delta A)(I + E) = D_1(C + \delta C)D_2(I + E). \tag{5.14}$$

If $I + E = (I + F)S$ is the LQ factorization, then $\|F\|_F \leq \sqrt{8}\|E\|_F + \sqrt{2}\|E\|_F^2$, provided that $\|E\|_F \leq 1/5$; see [17]. Further, $\|I - S\|_2 \leq \|E\|_2 + \|F\|_2$, and

$$A + \Delta A = D_1(C + \delta C)(I + F_1)D_2 S, \quad F_1 = D_2 F D_2^{-1}, \quad \|F_1\|_F \leq q\|F\|_F. \tag{5.15}$$

If we let $\Delta C = \delta C + CF_1 + \delta CF_1$, then using (5.8) we obtain (5.13). ☐

For SVD perturbation under this backward error with two-sided scaling we refer to [33], [9], [14], [10], [15], [24].

**5.4. Forward relative errors in the computed SVD.** Since the Jacobi SVD algorithm has columnwise small backward error, the condition number for the errors in the singular values of $A+\delta A = (I+\delta AA^\dagger)A$ is up to a $\sqrt{n}$ factor $\min_{D=\mathrm{diag}} \kappa_2(AD)$. This is in sharp contrast with bidiagonalization-based methods where the backward error has no columnwise structure and the condition number is $\kappa_2(A)$.

THEOREM 5.5. *Consider $A \in \mathbb{R}^{m\times n}$ with the SVD $A = U\binom{\Sigma}{0}V^T$ and singular values $\sigma_1 \geq \cdots \geq \sigma_n > 0$. Let $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$ be the singular values of the perturbed matrix $\tilde{A} = A + \delta A = (I + \Gamma)A$, $\Gamma = \delta AA^\dagger$, and let $\|\Gamma\|_2 < 1$.*

(i) *It holds that*

$$\max_{j=1:n} \frac{|\tilde{\sigma}_j - \sigma_j|}{\sqrt{\tilde{\sigma}_j\sigma_j}} \leq \|Sym(\Gamma)\|_2 + \frac{1}{2}\frac{\|\Gamma\|_2^2}{1 - \|\Gamma\|_2} \leq \|\Gamma\|_2 + O(\|\Gamma\|_2^2). \tag{5.16}$$

(ii) *Let $I+\Xi = \mathrm{diag}(\|(I+\Gamma)U(:,i)\|_2)_{i=1}^n$, $\breve{U} = (I+\Gamma)U(I+\Xi)^{-1}$, $\breve{U}^T\breve{U} = I+\Omega$, and $\hat{\Omega} = \Omega(1:n, 1:n)$. Let the singular values of $\tilde{A}$ be written with multiplicities as*

$$\tilde{\sigma}_1 = \cdots = \tilde{\sigma}_{\tilde{s}_1} > \tilde{\sigma}_{\tilde{s}_1+1} = \cdots = \tilde{\sigma}_{\tilde{s}_2} > \cdots > \tilde{\sigma}_{\tilde{s}_{\tilde{\ell}-1}+1} = \cdots = \tilde{\sigma}_{\tilde{s}_{\tilde{\ell}}}, \quad \tilde{s}_{\tilde{\ell}} = n, \quad \tilde{s}_0 \equiv 0,$$

*and let the relative gaps be defined by $\tilde{\gamma}_i = \min_{j\neq i} \frac{|\tilde{\sigma}_{\tilde{s}_i}^2 - \tilde{\sigma}_{\tilde{s}_j}^2|}{\tilde{\sigma}_{\tilde{s}_i}^2 + \tilde{\sigma}_{\tilde{s}_j}^2}$, $i = 1, \ldots, \tilde{\ell}$, $\tilde{\gamma} = \min_i \tilde{\gamma}_i$. If $\|\hat{\Omega}\|_2 < \tilde{\gamma}/3$, then for all $i$ and $\breve{\sigma}_j = \sigma_j\|(I + \Gamma)U(:,j)\|_2 = \sigma_j(1 + \Xi_{jj})$*

$$\sqrt{\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \left|\frac{\tilde{\sigma}_{\tilde{s}_i} - \breve{\sigma}_j}{\breve{\sigma}_j}\right|^2} \leq \sqrt{\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \left|1 - \frac{\tilde{\sigma}_{\tilde{s}_i}^2}{\breve{\sigma}_j^2}\right|^2} \leq \frac{2}{\tilde{\gamma}_i}\|\hat{\Omega}\|_2^2.$$

*In particular,* $\max_{j=1:n} \frac{|\tilde{\sigma}_j - \breve{\sigma}_j|}{\breve{\sigma}_j} \leq \frac{2}{\tilde{\gamma}} \|\hat{\Omega}\|_2^2$.

(iii) *For columnwise small* $\delta A$, $\|\Gamma\|_F \leq \sqrt{n} \max_{i=1:n}(\|\delta A(:,i)\|_2 / \|A(:,i)\|_2) \|A_c^{\dagger}\|_2$, *where* $A_c$ *is obtained by scaling the columns of* $A$ *to have unit norm.*

*Proof.* Since $I + \Gamma$ is nonsingular, we can use [27] and relation $(I+\Gamma)^{-1} = (I - \Gamma) + \Gamma^2 (I+\Gamma)^{-1}$ to conclude that

$$\max_{1 \leq j \leq n} \frac{|\tilde{\sigma}_j - \sigma_j|}{\sqrt{\tilde{\sigma}_j \sigma_j}} \leq \frac{1}{2} \|(I+\Gamma)^{-1} - (I+\Gamma)^T\|_2 = \frac{1}{2} \| - 2 Sym(\Gamma) + \Gamma^2 (I+\Gamma)^{-1}\|_2.$$

Relation (5.16) follows using the fact that $\|\Gamma\|_2 < 1$. Write

$$\tilde{A} = (I+\Gamma) U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T = \breve{U}(I + \Xi) \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^T, \tag{5.17}$$

where $(I+\Gamma)U = \breve{U}(I+\Xi)$ with diagonal matrix $\Xi$ determined so that $\breve{U}$ has unit columns. Obviously, $|\Xi_{ii}| \leq \|\Gamma U(:,i)\|_2$ for all $i$, and $\|\Xi\|_2 \leq \|\Gamma\|_2$. Write $\tilde{A}$ as

$$\tilde{A} = \breve{U} \begin{pmatrix} \breve{\Sigma} \\ 0 \end{pmatrix} V^T, \quad \begin{pmatrix} \breve{\Sigma} \\ 0 \end{pmatrix} = (I + \Xi) \begin{pmatrix} \Sigma \\ 0 \end{pmatrix}, \quad \breve{\Sigma} = \mathrm{diag}(\breve{\sigma}_j)_{j=1}^n, \tag{5.18}$$

and note that $\breve{U}^T \breve{U} = I + \Omega$ with $\Omega_{ii} = 0$ for all $i$. Now,

$$\tilde{A}^T \tilde{A} = V \begin{pmatrix} \breve{\Sigma} & 0 \end{pmatrix} (I + \Omega) \begin{pmatrix} \breve{\Sigma} \\ 0 \end{pmatrix} V^T = V \breve{\Sigma} (I_n + \hat{\Omega}) \breve{\Sigma} V^T, \tag{5.19}$$

where $\hat{\Omega} = \Omega(1:n, 1:n)$. Using the orthogonal similarity in the last relation, we can compare the eigenvalues of $\tilde{A}^T \tilde{A}$ and the corresponding eigenvalues of the matrix $M \equiv \breve{\Sigma}(I_n + \hat{\Omega})\breve{\Sigma}$. A second look at the relations (5.17)–(5.19) reveals the transformation of the multiplicative perturbation $I + \Gamma$ of $A$ into the nonorthogonality of the left singular vector matrix $U$ and then the splitting of the nonorthogonality of $(I+\Gamma)U$ into the column length changes and angle changes. The changes of the unit lengths of the columns of $U$ are then taken as perturbation of $\Sigma$, thus defining $\breve{\Sigma}$.

Note that the matrix $M$ is $\|\hat{\Omega}\|_2$—scaled diagonally dominant (s.d.d.) [2] with eigenvalues $\tilde{\sigma}_1^2 \geq \cdots \geq \tilde{\sigma}_n^2$ and diagonal entries $\breve{\sigma}_1^2 \geq \cdots \geq \breve{\sigma}_n^2$. Using [22, Corollary 3.2] we conclude that

$$\sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \left| 1 - \frac{\tilde{\sigma}_{\tilde{s}_i}^2}{(\breve{\sigma}_j)^2} \right|^2 + \sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \sum_{k=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \hat{\Omega}_{jk}^2$$
$$\leq \frac{4}{\tilde{\gamma}_i^2} \left( \sum_{j=\tilde{s}_{i-1}+1}^{\tilde{s}_i} \Big( \sum_{k=1}^{\tilde{s}_{i-1}} \hat{\Omega}_{jk}^2 + \sum_{k=\tilde{s}_i+1}^{n} \hat{\Omega}_{jk}^2 \Big) \right)^2 . \qquad \square$$

*Remark* 5.1. Consider the right-handed Jacobi SVD algorithm on $X \in \mathbb{R}^{n \times n}$. Let $\tilde{X}_\infty \equiv \tilde{X}^{(\overline{k})} = (X + \delta X)\hat{V}$ be the computed matrix and $\tilde{X}_\infty + \delta \tilde{X}_\infty = \tilde{U}\tilde{\Sigma}$ as in relation (4.3). Let $\max_{i \neq j} \left|(\tilde{U}^T \tilde{U})_{ij}\right| \leq \tau$, $\max_i \left|1 - \|\tilde{U}(:,i)\|_2\right| \leq \nu$. We wish to know how the sizes of $\tau$ and $\nu$ influence the relative distance between the $\tilde{\sigma}_i = \tilde{\Sigma}_{ii}$ and the corresponding exact singular value $\hat{\sigma}_i$ of $\tilde{U}\tilde{\Sigma}$. As in the proof of Theorem 5.5, we split the perturbation (the departure from orthogonality of $\tilde{U}$) into two parts. Let $\tilde{U} = \breve{U}(I + \Xi)$, where $\breve{U}$ has unit columns and $\Xi$ is a diagonal matrix with

$\|\Xi\|_2 \le \nu$. Write $\tilde{U}\tilde{\Sigma}$ as $\breve{U}\breve{\Sigma}$, where $\breve{\Sigma}$ is the diagonal matrix with diagonal entries $\breve{\sigma}_i = \tilde{\sigma}_i(1 + \Xi_{ii})$. Note that $\nu$ can be as small as $O(\varepsilon)$ with the cost of doubly accumulated dot products, and $O(n\varepsilon)$ if no extra precision is used. The potentially larger and harder to control value $\tau$ enters the estimate quadratically, and that opens a possibility for better stopping criteria. As in Theorem 5.5, we note that $\breve{\Sigma}\breve{U}^T\breve{U}\breve{\Sigma}$ has diagonal entries $\breve{\sigma}_i^2$ and eigenvalues $\hat{\sigma}_i^2$, $i = 1, \ldots, n$. Further, $\Omega = \breve{U}^T\breve{U} - I$ satisfies $\max_{ij} |\Omega_{ij}| \le \tau/(1-\nu)^2$. Let $\|\Omega\|_2 < \hat{\gamma}/3$, where the gap $\hat{\gamma}$ between the $\hat{\sigma}_i^2$'s is analogous to $\tilde{\gamma}$ in Theorem 5.5. Then, if $\hat{k}_i$ is the multiplicity of $\hat{\sigma}_i$, it holds that

$$\frac{|\hat{\sigma}_i - \breve{\sigma}_i|}{\breve{\sigma}_i} \le \frac{2}{\hat{\gamma}_i}\hat{k}_i(n - \hat{k}_i)\frac{\tau^2}{(1-\nu)^4} \le \frac{1}{\hat{\gamma}_i}\frac{n^2\tau^2}{2(1-\nu)^4}. \tag{5.20}$$

*Example* 5.1. We illustrate the application of the relation (5.20) in stopping the Jacobi SVD algorithm. Let $\varepsilon \approx 2.22 \cdot 10^{-16}$, $n = 1000$, and $\tau = 10^{-8}$. Since we do not have the $\hat{\sigma}_i$'s, the relative gaps will be estimated using the computed $\tilde{\sigma}_i$'s. Let $\tilde{U}^T\tilde{U} = I + \tilde{\Omega}$. Then $\|\tilde{\Omega}\|_F \le \omega \equiv \sqrt{n(n-1)\tau^2 + n\nu^2} < 9.9950 \cdot 10^{-6}$ and

$$\max_{i=1:n} \frac{|\hat{\sigma}_i - \tilde{\sigma}_i|}{\sqrt{\hat{\sigma}_i\tilde{\sigma}_i}} \le \|(I + \tilde{\Omega})^{-1/2} - (I + \tilde{\Omega})^{1/2}\|_2 \le \omega_1 \equiv \frac{\omega}{\sqrt{1-\omega}} < 9.9951 \cdot 10^{-6}.$$

From this we conclude that for all $i$

$$\frac{|\hat{\sigma}_i - \tilde{\sigma}_i|}{\min\{\hat{\sigma}_i, \tilde{\sigma}_i\}} \le \omega_2 \equiv \frac{\omega_1}{1-\omega_1} < 9.996 \cdot 10^{-6}, \quad \frac{|\hat{\sigma}_i - \tilde{\sigma}_i|}{\hat{\sigma}_i + \tilde{\sigma}_i} \le \frac{\omega_1}{2} < 4.998 \cdot 10^{-6}.$$

Suppose that we have $n$ different values $\tilde{\sigma}_1 > \cdots > \tilde{\sigma}_n > 0$ and that they are well separated relative to their uncertainty in approximating the $\hat{\sigma}_i$'s, i.e., let

$$\max_{i \ne j} \frac{|\tilde{\sigma}_i - \tilde{\sigma}_j|}{\tilde{\sigma}_i + \tilde{\sigma}_j} > 5\omega > 4.997 \cdot 10^{-5}. \text{ Then } \tilde{\gamma}_i \equiv \min_{j \ne i} \frac{|\tilde{\sigma}_i^2 - \tilde{\sigma}_j^2|}{\tilde{\sigma}_i^2 + \tilde{\sigma}_j^2} > 5\omega.$$

Since the $\hat{\sigma}_i$'s are $O(\omega)$ close to the $\tilde{\sigma}_i$'s, we know that the $\hat{\sigma}_i$'s are simple and that $\hat{\gamma}_i \equiv \min_{j \ne i} \frac{|\hat{\sigma}_i^2 - \hat{\sigma}_j^2|}{\hat{\sigma}_i^2 + \hat{\sigma}_j^2} \ge \tilde{\gamma}_i\big(1 - \frac{\omega_2}{5\omega}\big)\frac{1-\omega_2}{(1+\omega_2)^2} > 0.7999\tilde{\gamma}_i > 3.999\omega > 3\|\Omega\|_2$. Since $\breve{\sigma}_i = \tilde{\sigma}_i(1 + O(10^{-13}))$, we have $\breve{\sigma}_1 > \cdots > \breve{\sigma}_i > \breve{\sigma}_{i+1} > \cdots > \breve{\sigma}_n > 0$. We can now apply the quadratic bound, which yields for each $i$

$$\frac{|\hat{\sigma}_i - \breve{\sigma}_i|}{\breve{\sigma}_i} \le \frac{2}{0.7999}\frac{1}{\tilde{\gamma}_i}(n-1)\frac{\tau^2}{(1-\nu)^4} \le \frac{1}{\tilde{\gamma}_i}2.498 \cdot 10^{-13}. \tag{5.21}$$

Thus, if for instance $\tilde{\gamma}_i > 10^{-3}$, we can claim that $\tilde{\sigma}_i$ coincides with the corresponding $\hat{\sigma}_i$ to about ten decimal places, which actually doubles the previous number by about five known correct digits.

**5.5. Accuracy of the singular vectors.** The structure of the backward error in our algorithm is such that we can use well-developed and sharp perturbation theory [19], [28]. Our starting point is the relation (5.3) in Proposition 5.1, which is the SVD of $A+\mathcal{E}$ with the computed singular values in diagonal $\tilde{\Sigma}$, and exactly orthogonal matrices $\hat{Q}$, $\hat{V}_x$, $\hat{U}$ which are close to the corresponding computed approximations $\tilde{Q}$, $\overline{V}_x$, $\tilde{U}_x$, respectively. We first deal with the singular vector perturbations in the case of simple well-separated singular values. If $\sigma_1 \ge \cdots \ge \sigma_n$ are the singular values of $A = U\big(\begin{smallmatrix}\Sigma \\ 0\end{smallmatrix}\big)V^T$, then the relative separation is defined as $\rho_i = \min\big\{2, \min_{j \ne i} \frac{|\sigma_j - \sigma_i|}{\sigma_i}\big\}$, $i = 1, \ldots, n$. If the singular values are simple, then each $\rho_i$ is positive and the singular vectors define one-dimensional singular subspaces. If the perturbed matrix also has only simple singular values, then we can use the angles between the original and the perturbed subspaces as a natural error measure. Let $\theta_i$ and $\vartheta_i$ denote the error angles in the $i$th left and right singular vector, respectively. In the case of the perturbation from relation (5.3), $\theta_i = \angle(U(:,i), \hat{U}_a(:,i))$, $\vartheta_i = \angle(V(:,i), \hat{V}_a(:,i))$.

Proposition 5.6. *Let* $A = U\binom{\Sigma}{0}V^T$ *be the SVD of* $A$ *and let* (5.3) *be the SVD of a perturbed matrix with* $\|\mathcal{E}(:,i)\|_2 \leq \hat{\eta}\|A(:,i)\|_2$, $i = 1, \ldots, n$ *(cf. Proposition* 5.1*). Let* $\Phi = \mathcal{E}A^\dagger$, $\phi = \|\Phi + \Phi^T + \Phi\Phi^T\|_2$, $\phi \leq 2\|Sym(\Phi)\|_2 + \|\Phi\|_2^2$. *If* $\phi < \rho_i$, *then*

$$\max\{\sin\theta_i, \sin\vartheta_i\} \leq \sqrt{2}\left\{\frac{\xi}{\rho_i - \phi} + \|\Phi\|_2\right\}, \tag{5.22}$$

*where* $\xi \leq 2\|Sym(\Phi)\|_2 + O(\|\Phi\|_2^2)$ *and* $\|\Phi\|_2 \leq \sqrt{n}\hat{\eta}\|A_c^\dagger\|_2$.

*Proof.* We obtain the proof by applying [19, Theorem 3.3]. □

Application of the above estimates to the actually computed matrices $\tilde{U}_a$, $\tilde{V}_a$ follows by combining Propositions 5.6 and 5.2, since the angles $\angle(\tilde{U}_a(:,i), \hat{U}_a(:,i))$ and $\angle(\tilde{V}_a(:,i), \hat{V}_a(:,i))$ are small, with bounds sharper than those in (5.22).

In cases of clustered or multiple singular values, the singular vectors are not the right objects to be approximated numerically. Instead, we try to compute well-defined singular subspaces, belonging to multiple or tightly grouped singular values. The structure of the backward perturbation in the Jacobi SVD algorithm fits nicely into the perturbation estimates. For the sake of simplicity, we will give only one perturbation result, following [28]. Other interesting bounds can be derived from the fact that $A + \mathcal{E} = (I + \Phi)A$, where $\|\Phi\|$ is independent of the column scaling of $A$.

Proposition 5.7. *Let* $\Sigma = \Sigma_1 \oplus \Sigma_2$, $\tilde{\Sigma} = \tilde{\Sigma}_1 \oplus \tilde{\Sigma}_2$ *with* $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$, $\Sigma_2 = \mathrm{diag}(\sigma_{k+1}, \ldots, \sigma_n)$, $\tilde{\Sigma}_1 = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_k)$, $\tilde{\Sigma}_2 = \mathrm{diag}(\tilde{\sigma}_{k+1}, \ldots, \tilde{\sigma}_n)$. *Let* $\varrho = \min_{i=1:k;j=1:n-k} |\sigma_i - \tilde{\sigma}_{k+j}|/\sqrt{\sigma_i^2 + \tilde{\sigma}_{k+j}^2}$. *In the rectangular case,* $m > n$, *replace* $\varrho$ *with* $\min\{\varrho, 1\}$. *Let* $\mathcal{U}_1$, $\hat{\mathcal{U}}_1$, $\mathcal{V}_1$, $\hat{\mathcal{V}}_1$ *be the subspaces spanned by the columns of* $U_1 \equiv U(:,1:k)$, $\hat{U}_a(:,1:k)$, $V(:,1:k)$, $\hat{V}_a(:,1:k)$, *respectively. If* $\varrho > 0$, *then*

$$\left\|\begin{pmatrix} \|\sin\Theta(\mathcal{U}_1, \hat{\mathcal{U}}_1)\|_F \\ \|\sin\Theta(\mathcal{V}_1, \hat{\mathcal{V}}_1)\|_F \end{pmatrix}\right\|_F \leq \frac{\sqrt{\|\Phi^T U_1\|_F^2 + \|-\Phi U_1 + \Phi^2(I-\Phi)^{-1}U_1\|_F^2}}{\varrho}. \tag{5.23}$$

*Thus, the error angles are bounded by* $O(\|\Phi\|_F/\varrho)$.

This concludes the first part of our report. We have defined the global structure of a new preconditioned Jacobi SVD algorithm, which uses pivoted QR factorization as the preconditioner and reduces the computation to the SVD of structured triangular matrices. We have shown that the new algorithm computes the SVD with columnwise small backward error and with condition number independent of column scaling. Reliable implementation of the preconditioner is given in [16]. The new implementation of the Jacobi SVD on triangular matrices and the results of numerical testing are presented in [18], where we show that the new method can reach the efficiency of less accurate bidiagonalization-based methods (SGESVD and SGESDD from LAPACK). The speedup over the equally accurate standard one-sided Jacobi SVD can be a factor of ten or more.

**Acknowledgments.** The authors acknowledge generous support by the Volkswagen Science Foundation and the Croatian Ministry of Science and Technology. They are also indebted to P. Arbenz (Zürich), J. Barlow (State College), J. Demmel (Berkeley), F. Dopico (Madrid), V. Hari (Zagreb), W. Kahan (Berkeley), J. Moro

(Madrid), B. Parlett (Berkeley), and I. Slapničar (Split) for their comments, criticisms, and many fruitful discussions. Special thanks go to the anonymous referees for their substantial and constructive suggestions.

## REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenny, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1995.

[2] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.

[3] R. Bhatia, *Matrix Analysis*, Grad. Texts in Math., Springer-Verlag, New York, 1997.

[4] C. H. Bischof and G. Quintana-Orti, *Algorithm* 782: *Codes for rank–revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, 24 (1998), pp. 254–257.

[5] C. H. Bischof and G. Quintana-Orti, *Computing rank-revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, 24 (1998), pp. 226–253.

[6] P. A. Businger and G. H. Golub, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.

[7] S. Chandrasekaran and I. C. F. Ipsen, *On rank-revealing factorisations*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 592–622.

[8] S. Chandrasekaran and I. C. F. Ipsen, *Analysis of a QR algorithm for computing singular values*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 520–535.

[9] A. J. Cox and N. J. Higham, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, Proceedings of the 17th Dundee Biennial Conference, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., Pitman Res. Notes in Math. Ser. 380, Longman, Harlow, UK, 1998, pp. 57–73.

[10] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.

[11] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.

[12] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[13] Z. Drmač, *Computing the Singular and the Generalized Singular Values*, Ph.D. thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Germany, 1994.

[14] Z. Drmač, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.

[15] Z. Drmač, *On principal angles between subspaces of Euclidean space*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 173–194.

[16] Z. Drmač and Z. Bujanović, *On the failure of rank revealing QR factorization software—a case study*, ACM Trans. Math. Software, to appear.

[17] Z. Drmač, M. Omladič, and K. Veselić, *On the perturbation of the Cholesky factorization*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1319–1332.

[18] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm.* II, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1343–1362.

[19] S. C. Eisenstat and I. C. F. Ipsen, *Relative perturbation techniques for singular value problems*, SIAM J. Numer. Anal., 32 (1995), pp. 1972–1988.

[20] K. V. Fernando and B. N. Parlett, *Implicit Cholesky algorithms for singular values and vectors of triangular matrices*, Numer. Linear Algebra Appl., 2 (1995), pp. 507–531.

[21] M. Gu and S. C. Eisenstat, *A divide-and-conquer algorithm for the bidiagonal SVD*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 79–92.

[22] V. Hari and Z. Drmač, *On scaled almost-diagonal Hermitian matrix pairs*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1000–1012.

[23] M. R. Hestenes, *Inversion of matrices by biorthogonalization and related results*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 51–90.

[24] N. J. Higham, *QR factorization with complete pivoting and accurate computation of the SVD*, Linear Algebra Appl., 309 (2000), pp. 153–174.

[25] C. G. J. Jacobi, *Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Crelle's Journal für reine und angew. Math., 30 (1846), pp. 51–95.

[26] W. Kahan, *Numerical linear algebra*, Canad. Math. Bull., 9 (1965), pp. 757–801.

[27] R.-C. Li, *Relative perturbation theory:* I. *Eigenvalue and singular value variations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 956–982.

[28] R.-C. Li, *Relative perturbation theory:* II. *Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492.

[29] R. Mathias and G. W. Stewart, *A block QR algorithm for singular value decomposition*, Linear Algebra Appl., 182 (1993), pp. 91–100.

[30] M. J. D. Powell and J. K. Reid, *On applying Householder transformations to linear least squares problems*, in Information Processing 68 (Proc. IFIP Congress, Edinburgh, 1968), North-Holland, Amsterdam, 1969, pp. 122–126.

[31] H. Rutishauser, *Vorlesungen über numerische Mathematik*, Band 2, Differentialgleichungen und Eigenwertprobleme, Birkhäuser Verlag, Basel, Stuttgart, 1976. Lehrbücher und Monographien aus dem Gebiete der exakten Wissenschaften, Math. Reihe, Band 57.

[32] G. W. Stewart, *The QLP Approximation to the Singular Value Decomposition*, Tech. report TR-97-75, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1997.

[33] J.-G. Sun, *Componentwise perturbation bounds for some matrix decompositions*, BIT, 32 (1992), pp. 702–714.

[34] K. Veselić and V. Hari, *A note on a one-sided Jacobi algorithm*, Numer. Math., 56 (1989), pp. 627–633.

SIAM J. Matrix Anal. Appl.
Vol. 29, No. 4, pp. 1343–1362

# NEW FAST AND ACCURATE JACOBI SVD ALGORITHM. II*

ZLATKO DRMAČ† AND KREŠIMIR VESELIĆ‡

**Abstract.** This paper presents a new one-sided Jacobi SVD algorithm for triangular matrices computed by revealing QR factorizations. If used in the preconditioned Jacobi SVD algorithm, described in part one of this paper, it delivers superior performance leading to the currently fastest method for computing SVD decomposition with high relative accuracy. Furthermore, the efficiency of the new algorithm is comparable to the less accurate bidiagonalization-based methods. The paper also discusses underflow issues in floating point implementation and shows how to use perturbation theory to fix the imperfectness of the machine arithmetic.

**Key words.** Jacobi method, singular value decomposition, eigenvalues

**AMS subject classifications.** 15A09, 15A12, 15A18, 15A23, 65F15, 65F22, 65F35

**DOI.** 10.1137/05063920X

## 1. Introduction.

Jacobi iteration is one of the time-honored methods for computing the spectral decomposition $H = V\Lambda V^T$ of a real symmetric matrix $H$. The early discovery in 1846 is certainly due to the simplicity and the elegance of the method as well as to the geniality of C. G. J. Jacobi, who called it "ein leichtes Verfahren" and applied it to compute the secular perturbations of the planets. Jacobi's original article [25] is a masterpiece of applied mathematics and may even today be read with profit by both students and scientists. The simplicity of the Jacobi method is not only theoretical but also computational, and in this respect it may well be compared with Gaussian elimination. Thus, with coming of automatic computation, the Jacobi method was soon rediscovered by Goldstine, Murray, and von Neumann [17], who provided the first detailed implementation and error analysis.

In our recent work [14] we introduced a preconditioner for the Hestenes variant [24] of the Jacobi method for SVD computation of general matrices. We have shown that rank revealing QR factorization can serve as a versatile preconditioner which enables efficient execution of Jacobi iterations on the triangular factor. The idea of QR iterations as preconditioner for SVD computation is well known (see [33], [27], [15]), but thus far it has not been fully exploited in the context of the Jacobi method. It is both simple and powerful: If $AP = Q(R^T \; 0)^T$ is the Businger–Golub QR factorization of $A$, then the Hestenes one-sided Jacobi algorithm applied to $X = R^T$ converges much faster than if applied to $A$. (If $R$ is singular, then the second QR factorization $R^T P_1 = Q_1 \left( R_1^T \quad 0 \right)^T$ provides nonsingular $X = R_1^T$.) In [14] Jacobi iterations on triangular matrices are used as a *black-box* procedure: starting with $X^{(0)} = X$, the sequence $X^{(k+1)} = X^{(k)} V^{(k)}$ converges to $X_\infty = U\Sigma$ and the product of Jacobi rotations $V^{(0)}V^{(1)}\cdots$ converges to $V$. The SVD of $X$ is $X = U\Sigma V^T$, where the matrix

---

*Received by the editors August 30, 2005; accepted for publication (in revised form) by M. Chu June 5, 2007; published electronically January 4, 2008. This work was supported by the Volkswagen–Stiftung grant *Designing Highly Accurate Algorithms for Eigenvalue and Singular Value Decompositions.*

http://www.siam.org/journals/simax/29-4/63920.html

†Department of Mathematics, University of Zagreb, Bijenička 30, 10000 Zagreb, Croatia (drmac@math.hr). The work of this author was supported by the Croatian Ministry of Science and Technology under grant 0037120 (*Numerical Analysis and Matrix Theory*).

‡Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Postfach 940, D-58084 Hagen, Germany (kresimir.veselic@FernUni-Hagen.de).

$V$ is obtained not from the accumulated product of Jacobi rotations but rather in an a posteriori manner, using the relation $V = X^{-1}X_{\infty}$. Assembling the SVD of $A$ from the SVD of $X$ is straightforward.

In this report we unwrap the black box and present a new one-sided Jacobi SVD method for triangular matrices. A new pivot strategy is introduced in section 2. We use the triangular structure to reduce the flop count and memory traffic. At the same time, faster convergence is achieved using the knowledge of the asymptotic behavior of Jacobi iterations. We also use the structure of the SVD of triangular matrices, obtained from the theory of symmetric quasi-definite matrices. A new ordering of rotations is also designed to improve the use of fast cache memory. Underflow problems in floating point implementation of the algorithm are solved using perturbation theory in section 3. Numerical testing of the new preconditioned Jacobi SVD algorithm (cf. [14, Algorithm 4] with the new triangular SVD method from this paper) is presented in section 4. The results presented in section 4.3 carry the main message of [8], [14], and this paper: *Our new Jacobi SVD algorithm is more accurate than the bidiagonalization-based QR (SGESVD) and divide-and-conquer (SGESDD) algorithms from LAPACK* [1]. *Moreover, the new algorithm can compute the SVD faster than SGESVD, and it is not much slower than SGESDD.* Concluding remarks and discussion of future work are given in section 5.

## 2. One-sided Jacobi SVD on $n \times n$ preconditioned triangular matrices.

The Jacobi transformation $X^{(k+1)} = X^{(k)}V^{(k)}$ transforms pivot columns $p_k$, $q_k$ chosen by pivot strategy (ordering) $k \mapsto (p_k, q_k)$. An example is the row-cyclic strategy, which is periodic, and in one full sweep of $n(n-1)/2$ rotations it rotates at the pivot positions $(1,2), (1,3), \ldots, (1,n);\ (2,3), \ldots, (2,n);\ (3,4), \ldots, (3,n); \ldots, (n-2,n);\ (n-1,n)$. The convergence of $X^{(k)}$ is studied in terms of the convergence of $H^{(k)} = (X^{(k)})^T X^{(k)}$ towards the diagonal form, and its rate is usually measured using the off-norm, $\boldsymbol{\Omega}(H^{(k)}) = \|H^{(k)} - \mathrm{diag}(H^{(k)})\|_F$. In the case of the row-cyclic strategy the convergence is asymptotically quadratic [20]: If $\boldsymbol{\Omega}(H^{(0)})$ is sufficiently small and if the diagonal entries of $H^{(0)}$ are sorted, then $\boldsymbol{\Omega}(H^{(n(n-1)/2)}) \leq \mathrm{const} \cdot \boldsymbol{\Omega}(H^{(0)})^2$.

In practice, the Jacobi rotation $V^{(k)}$ is executed only if the cosine of the angle between $X^{(k)}(:,p_k)$ and $X^{(k)}(:,q_k)$ is greater than a tolerance which is usually $n$ times the round-off $\boldsymbol{\varepsilon}$. Otherwise, the rotation is skipped. If $n(n-1)/2$ consecutive rotations with all possible pivot pairs are skipped, i.e.,

$$\max_{i \neq j} \frac{|X(:,i)^T X(:,j)|}{\|X(:,i)\|_2 \|X(:,j)\|_2} \leq n\boldsymbol{\varepsilon}, \tag{2.1}$$

then the iterations are stopped and numerical convergence is declared. Demmel and Veselić [8] showed that (2.1) is important for high relative accuracy. If the floating point Jacobi rotation is implemented as in [11], then the procedure can compute the singular values in the full range of machine numbers.

Our new pivot strategy is based on the fact that the initial matrix $X = X^{(0)}$ is triangular, nonsingular, and with additional structure implied by the Businger–Golub column pivoting. In this section, we outline the key ingredients of the new approach.

### 2.1. SVD of lower triangular matrix.

As discussed in [14], the preconditioned Jacobi SVD algorithm computes a rank revealing QR factorization $AP = Q\binom{R}{0}$, and then it applies one-sided Jacobi SVD to $X = R^T$. (In some cases, it uses second QR factorization $R^T = Q_1 \begin{pmatrix} R_1^T & 0 \end{pmatrix}^T$ and then $X = R_1^T$.) The reason is that $RR^T$ is more diagonal than $R^T R$. In addition to the arguments in [14], we give an illustration by simple MATLAB example and show that the gap revealing property of the pivoted QR factorization (cf. [31], [32]) opens a connection to the theory of symmetric quasi-definite matrices.

*Example* 2.1. We generate in MATLAB $A \in \mathbb{R}^{50\times 50}$ with entries uniformly distributed over $[0,1]$, and compare the column norms of $A$ and $X = R^T$. Further, we compare the diagonality of the Gram matrices $H_s = R_c^T R_c$ and $M_s = X_c^T X_c$, where $R_c$, $X_c$ are obtained by column equilibration, e.g., $X_c = X\mathrm{diag}(\|X(:,i)\|_2^{-1})$. The results shown in Figure 2.1 strongly suggest that the Jacobi method should diagonalize $X^TX$ faster than $R^TR$. In the case of graded $A$ (column norms vary in length) the positive effect of using $X$ instead of $R$ is even stronger. (See the second row in Figure 2.1.) An analysis of perfect behavior of the column norms of $X$ can be found in [12].

It appears that the Jacobi iterations $X^{(k+1)} = X^{(k)}V^{(k)}$ in general work better if the initial $X$ is lower triangular. An explanation is given in the following theorem.

THEOREM 2.1. *Let $X \in \mathbb{R}^{n\times n}$ be a lower triangular matrix with the partition*

$$X = \begin{pmatrix} X_{11} & 0 \\ X_{21} & X_{22} \end{pmatrix}, X_{11} \in \mathbb{R}^{k\times k}, \quad \text{and let } \sigma_{\min}\left(\begin{pmatrix} X_{11} \\ X_{21}\end{pmatrix}\right) > \sigma_{\max}(X_{22}). \tag{2.2}$$

*Let the SVD of $X$ be given with the partition*

$$X = U\Sigma V^T = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix}\begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}\begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}^T. \tag{2.3}$$

*Then the matrix $V$ is more block diagonal than $U$ in the following sense:*

- (i) *In the Löwner partial order $V_{11}^TV_{11} \succ V_{21}^TV_{21}$, $V_{22}^TV_{22} \succ V_{12}^TV_{12}$. (For symmetric matrices $S_1$ and $S_2$, $S_1 \succ S_2$ if and only if $S_1 - S_2$ is positive definite.) As a consequence, $\sigma_{\min}(V_{11}) > 1/\sqrt{2}$ and $\sigma_{\min}(V_{22}) > 1/\sqrt{2}$.*
- (ii) *Let $\mathcal{U}_k$, $\mathcal{V}_k$, $\mathcal{I}_k$ be the subspaces spanned by the first $k$ columns of $U$, $V$, and the identity matrix $I$, respectively. If the angles $\psi_k$, $\theta_k$ are defined as $\psi_k = \angle(\mathcal{V}_k, \mathcal{I}_k)$, $\theta_k = \angle(\mathcal{U}_k, \mathcal{I}_k)$, then $\psi_k < \pi/4$ and*

$$\tan\psi_k \le \frac{\sigma_{k+1}}{\sigma_k}\tan\theta_k.$$

*Proof.* Consider the block partition of the cross product matrix $H = X^TX$,

$$X^TX = \begin{pmatrix} X_{11}^TX_{11} + X_{21}^TX_{21} & X_{21}^TX_{22} \\ X_{22}^TX_{21} & X_{22}^TX_{22} \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}.$$

The gap assumption (2.2) implies that for any shift $\xi \in \mathcal{S} = (\lambda_{\max}(H_{22}), \lambda_{\min}(H_{11}))$ both $H_{11} - \xi I$ and $\xi I - H_{22}$ are positive definite. Therefore, the matrix $H - \xi I$ is symmetric quasi-definite, and the matrix $V$ must have the special structure of the eigenvector matrix of quasi-definite matrices [16]. In particular, $V_{11}^TV_{11} \succ V_{21}^TV_{21}$, $V_{22}^TV_{22} \succ V_{12}^TV_{12}$, yielding $\sigma_{\min}(V_{11}) > 1/\sqrt{2}$, $\sigma_{\min}(V_{22}) > 1/\sqrt{2}$. From this and $X_{11}V_{11} = U_{11}\Sigma_1$ it follows that $U_{11}$ is nonsingular. Further, from the SVD of $X$ it follows that

$$V_{11}^{-1}V_{12} = -(V_{22}^{-1}V_{21})^T = \Sigma_1^{-1}U_{11}^{-1}U_{12}\Sigma_2 \quad \text{and} \quad \|V_{11}^{-1}V_{12}\|_2 \le \frac{\sigma_{k+1}}{\sigma_k}\|U_{11}^{-1}U_{12}\|_2.$$

Finally, from the CS decomposition of partitioned $V$ we have $\|V_{11}^{-1}V_{12}\|_2 = \tan\psi_k$. ☐

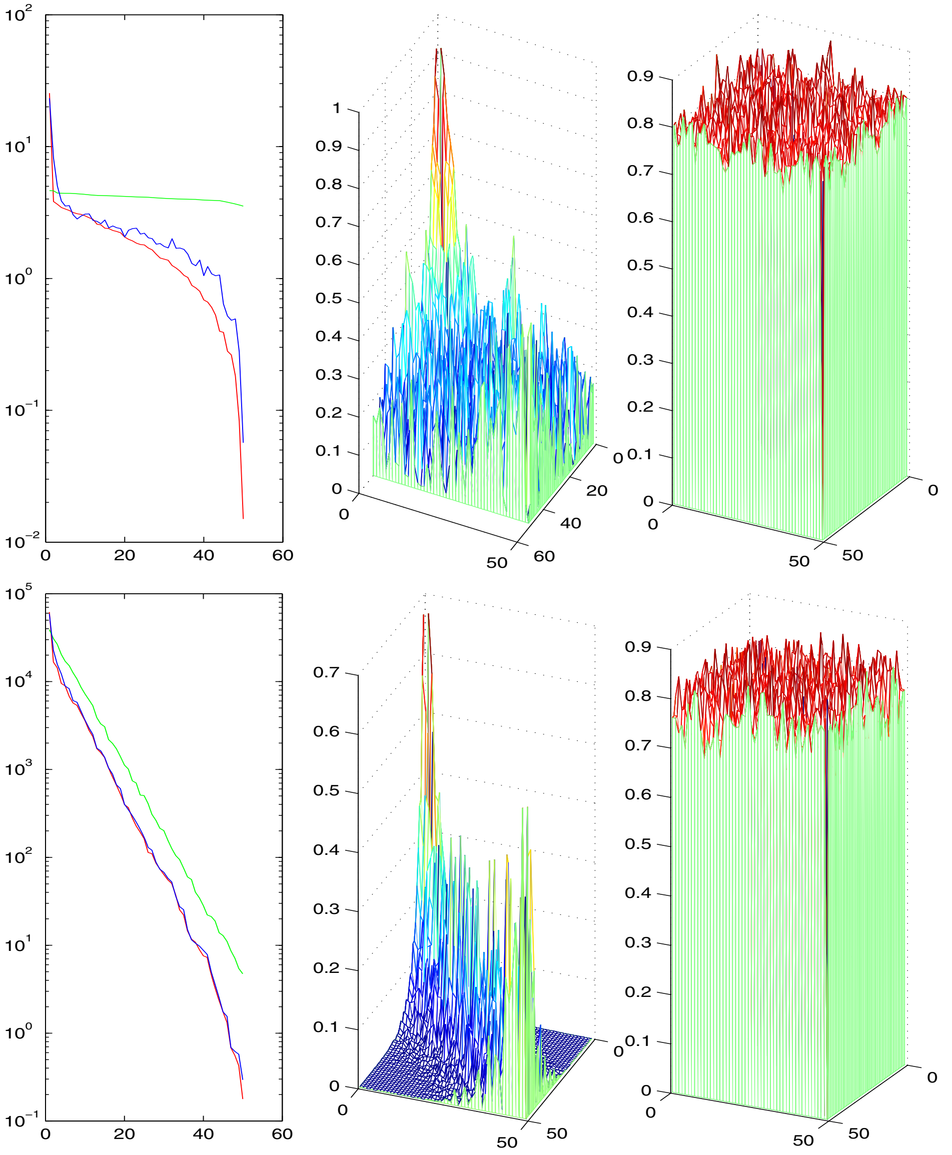


FIG. 2.1. *Example* 2.1*: In the first plot, the top line denotes* sorted *column norms of $A$, the middle line denotes the column norms of $X = R^T$, and lowest line denotes the singular values. The next two plots are obtained by meshz$(abs(M_s - \mathrm{diag}(M_s)))$ and meshz$(abs(H_s - \mathrm{diag}(H_s)))$, respectively. In the second row of the figure, the plots correspond to matrix $A$ with graded columns.*

*Remark* 2.1. Part (ii) of Theorem 2.1 is due to Chandrasekaran and Ipsen [6] but under the assumption that $X$ and one of the matrices $U_{11}$, $V_{11}$ is nonsingular. The contribution of (ii) is in establishing a connection between the separation condition (2.2) and the theory of symmetric quasi-definite matrices to conclude (i), which yields nonsingularity of $V_{11}$. Further, our "weak separation" condition (2.2) is weaker than the usual condition $\sigma_{\min}(X_{11}) > \sigma_{\max}(X_{22})$.

*Remark* 2.2. Comparing $X^TX$ with $XX^T = \begin{pmatrix} X_{11}X_{11}^T & X_{11}X_{21}^T \\ X_{21}X_{11}^T & X_{22}X_{22}^T + X_{21}X_{21}^T \end{pmatrix}$ we see that one important difference is that the monotonicity principle acts inside different diagonal blocks. Thus, it can increase or decrease the initial separation gap. Also note that in the case of "strong separation" $\sigma_{\min}(X_{11}) > \sigma_{\max}((X_{21}\ X_{22}))$, even the matrix $U$ has the property (i). However, property (ii) implies that $V$ is more block diagonal. Hence, we prefer $V$ to be the product of Jacobi rotations.

In the Jacobi SVD algorithm applied to $X = R^T$, the matrix $V$ is built as the accumulated product of Jacobi rotations. The structure of $V$ (more block diagonal than $U$) is an additional argument to apply Jacobi rotations to the columns of $X$.

**2.2. How to exploit the triangular form.** Classical pivot strategies in the one-sided Jacobi method are not designed to preserve any zero pattern of the input matrix. In fact, the incapability to preserve created zeros was the main reason for the poor performance, as compared with bidiagonalization-based methods.

However, if the initial matrix is triangular, quite a few rotations in the first sweep can use and partially preserve the zero structure. Let $X$ denote the array in the memory occupied by the iterates in the Jacobi SVD algorithm. Initially, $X$ is lower triangular; see (2.4). Let the columns of $X$ be partitioned into four blocks, $X_{[1]}$, $X_{[2]}$, $X_{[3]}$, $X_{[4]}$, of dimensions $n \times n_i$, respectively, where each $n_i$ is approximately $n/4$. Further, let $X^{[1]} = \big(X_{[1]}, X_{[2]}\big)$, $X^{[2]} = \big(X_{[3]}, X_{[4]}\big)$.

$$X = \left(\begin{array}{cc|cc||cc|cc}
\blacksquare & o & 0 & 0 & 0 & 0 & 0 & 0 \\
\blacksquare & \blacksquare & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\blacksquare & \blacksquare & \blacksquare & o & 0 & 0 & 0 & 0 \\
\blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 & 0 & 0 \\
\hline\hline
\blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & o & 0 & 0 \\
\blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 & 0 \\
\hline
\blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & o \\
\blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare
\end{array}\right) \equiv \big(X_{[1]}, X_{[2]}, X_{[3]}, X_{[4]}\big) \equiv \big(X^{[1]}, X^{[2]}\big). \tag{2.4}$$

We will say that rotations are applied in $X_{[i]}^{\circlearrowright}$ ($X_{\circlearrowright}^{[i]}$) if we implicitly transform $X_{[i]}^T X_{[i]}$ ($(X^{[i]})^T X^{[i]}$) by following one full sweep of some pivot strategy. Further, for two blocks $X_{[i]}$ and $X_{[j]}$ ($X^{[i]}$ and $X^{[j]}$) of $X$, rotating in $X_{[i]} \leftrightarrow X_{[j]}$ ($X^{[i]} \leftrightarrow X^{[j]}$) means choosing, in some order, all pivot pairs with the first pivot column from $X_{[i]}$ ($X^{[i]}$) and the second one from $X_{[j]}$ ($X^{[j]}$).

Consider the most natural greedy approach. Rotating in $X_{[4]}$ is efficient because all columns are, and remain during rotations, in a canonical subspace of dimension $n_4 \approx n/4$. Thus, only the submatrix $X_{[4]}(n_1 + n_2 + n_3 + 1 : n, 1 : n_4)$ is transformed. This reduces the operation count of rotations applied in $X_{[4]}^{\circlearrowright}$ by a factor of four. In the same way, transformations of the columns of $X_{[3]}$ by any strategy is computed in a subspace of dimension $n_3 + n_4 \approx n/2$. The same holds for the transformations of the columns of $X^{[2]}$. Repeated transformations of the columns of $X_{[3]}$ (and independently $X_{[4]}$) are still in a lower-dimensional subspace, and thus very efficient. Savings in the transformations of the columns of $X_{[2]}$ are modest, but not worthless. Note that this strategy in the first sweep transforms more often closer to the diagonal, which seems reasonable given the structure of the initial matrix (see Example 2.1).

DEFINITION 2.2. *The greedy triangular sweep for the partition* (2.4) *of lower triangular matrix is defined as the following ordering of Jacobi rotations:*

$$X_{[3]}^{\circlearrowright},\ X_{[4]}^{\circlearrowright},\ X_{[3]} \leftrightarrow X_{[4]},\ X_{[3]}^{\circlearrowright},\ X_{[4]}^{\circlearrowright};\ X_{[1]}^{\circlearrowright},\ X_{[2]}^{\circlearrowright},\ X_{[1]} \leftrightarrow X_{[2]},\ X_{[1]}^{\circlearrowright},\ X_{[2]}^{\circlearrowright};\ X^{[1]} \leftrightarrow X^{[2]}. \tag{2.5}$$

*In each bulk of rotations (*$X_{[i]}^{\circlearrowleft}$, $X_{[j]}\leftrightarrow X_{[k]}$, $X^{[j]}\leftrightarrow X^{[k]}$*) pivot ordering is arbitrary and implemented to transform only the nontrivial parts of the corresponding submatrices.*

Thus, in the greedy triangular sweep, the blocks of zeros denoted by "0" in (2.4) are at some point used to reduce the complexity (number of operations and memory traffic), as discussed above. This is particularly important because the first sweep is the busiest one. The positions denoted by "$o$" are treated as nonzero entries, i.e., they are not used to save operations.

This technique can be applied recursively by refining the partition (2.4).

**2.3. Cubic convergence.** Mascarenhas [26] observed that in the row-cyclic strategy the off-diagonal entries converge to zero at different rates and showed that by using special quasi-cyclic strategies the Jacobi method can attain cubic asymptotic convergence. Here the term quasi-cyclic refers to a modified row-cyclic strategy in which slowly convergent positions are visited more often. To motivate quasi-cyclic strategy, assume that $H = X^T X$ is almost diagonal, $\boldsymbol{\Omega}(H) = O(\epsilon)$, and introduce the following block partitions:

$$H = \left(\begin{array}{c||c} H^{[11]} & H^{[12]} \\ \hline\hline H^{[21]} & H^{[22]} \end{array}\right) = \left(\begin{array}{c|c||c|c} H_{[11]} & H_{[12]} & H_{[13]} & H_{[14]} \\ \hline H_{[21]} & H_{[22]} & H_{[23]} & H_{[24]} \\ \hline\hline H_{[31]} & H_{[32]} & H_{[33]} & H_{[34]} \\ \hline H_{[41]} & H_{[42]} & H_{[43]} & H_{[44]} \end{array}\right). \tag{2.6}$$

For simplicity, assume that the eigenvalues of $H$ are well separated. If the row-cyclic strategy is first applied to the diagonal blocks $H^{[11]}$ and then to $H^{[22]}$, their off-diagonal norms will be reduced from $O(\epsilon)$ to $O(\epsilon^2)$. Rotating in the row-cyclic fashion inside the block $H^{[12]}$ reduces its norm to the order of $O(\epsilon^3)$. Visiting the diagonal blocks $H^{[11]}$ and $H^{[22]}$ once more will reduce their off-norms to $O(\epsilon^4)$. Repeating this pattern recursively on a finer partition of $H$, we obtain the quasi-cyclic ordering which visits all pivot pairs from

$$H_{[33]}, H_{[44]}, H_{[34]}, H_{[33]}, H_{[44]}, H_{[11]}, H_{[22]}, H_{[12]}, H_{[11]}, H_{[22]}, H^{[12]}, \tag{2.7}$$

respectively. The pivot positions inside each block are visited using a row-cyclic strategy.

THEOREM 2.3 (see Rhee and Hari [28]). *Let the diagonal entries of $H$ in* (2.6) *be ordered from large to small, and let no two diagonal entries from different blocks $H_{[ii]}$, $H_{[jj]}$ be affiliated with the same eigenvalue. Let* $\boldsymbol{\delta} = \min_{\lambda_i\neq\lambda_j}|\lambda_i-\lambda_j|$ *and*

$$\Gamma_1 = \frac{\sqrt{\boldsymbol{\Omega}(H^{[11]})^2 + \boldsymbol{\Omega}(H^{[22]})^2}}{\boldsymbol{\delta}/3},\quad \Gamma_2 = \left(\frac{\|H^{[12]}\|_F}{\boldsymbol{\delta}/3}\right)^{2/3},\quad \Gamma \equiv \Gamma(H) = \max\{\Gamma_1, \Gamma_2\}.$$

*Let $H'$ be the matrix computed after the quasi-cycle* (2.7). *If* $\Gamma(H) < 1/4$, *then* $\Gamma(H') < (49/25)\Gamma(H)^3$. *Further, if* $\Gamma(H) = \Gamma_1 < 1/4$, *then* $\boldsymbol{\Omega}(H') \le (18/\boldsymbol{\delta}^2)\boldsymbol{\Omega}(H)^3$.

Rhee and Hari pointed out that the reduction of $\boldsymbol{\Omega}(H)$ is only quadratic if $\Gamma_2$ dominates $\Gamma_1$. One of the key points in our new preconditioned algorithm is that preconditioning makes the dominance of $\Gamma_2$ over $\Gamma_1$ very unlikely; see Example 2.1. Intuitively, it is then a reasonable strategy to take care of $\Gamma_1$ first and then follow the strategy (2.7) in its implicit form (2.5). We expect positive effects of the cubic convergence mechanism even in the first sweep, before the conditions for the cubic convergence are fulfilled.

Let us summarize the elements of the first sweep and the benefits of working on $X = R^T$, as compared to the classical application of the Hestenes Jacobi SVD on $A$:

$$X^TX = \left(\begin{array}{cccc||cccc}
\blacksquare & \boxplus & \boxminus & \boxminus & \otimes & \otimes & \otimes & \otimes \\
 & \blacksquare & \boxminus & \boxplus & \boxminus & \boxminus & \otimes & \otimes \\
 & & \blacksquare & \boxminus & \boxminus & \otimes & \otimes & \otimes \\
 & & & \blacksquare & \boxplus & \boxminus & \boxminus & \otimes \\
\hline\hline
 & & & & \blacksquare & \boxplus & \boxminus & \boxplus \\
 & & & & & \blacksquare & \boxplus & \boxminus \\
 & & & & & & \blacksquare & \boxplus \\
 & & & & & & & \blacksquare
\end{array}\right)$$

$\boxplus$ rotated
$\boxminus$ rotation skipped after test
$\otimes$ dot product test abandoned

FIG. 2.2. *Example of modified row-cyclic strategy: if two consecutive rotations in a row are skipped, then the remaining pivot positions in that row are not even tested against the threshold.*

(i) $R$ is computed efficiently by BLAS 3 operation; smaller dimension in the case $m > n$;

(ii) preconditioning effect ($X^TX$ closer to diagonal than $A^TA$; see Example 2.1);

(iii) a priori known structure of $X^TX$ and of Jacobi rotations (Theorem 2.1);

(iv) greedy sweep (2.5) exploits triangular structure to save flops and reduce memory traffic; it transforms more often where most needed, and at the same time it follows cubically convergent strategy (2.7).

**2.4. How to adapt to the nearly band structure.** Upon completion of the first sweep, the array $X$ is dense. However, $X$ is the result of the preconditioning by a rank revealing QR factorization, followed by a greedy sweep. Hence, $X$ must have structure that can be exploited. We expect that $H = X^TX$ is a scaled diagonally dominant (s.d.d.) matrix in the sense of [2] and that its off-diagonal mass is distributed close to the diagonal. This is a typical nonpathological situation. The pathological case occurs in the presence of multiple or tightly clustered singular values.

It is known that the convergence of Jacobi iterations is improved if the threshold for the rotation is set higher at the beginning of the process and then gradually reduced to the final level. Such a strategy is not suitable for the implicit Jacobi SVD algorithm because $O(n)$ flops are required to compute a single pivot element. Further, in the nonpathological case we may expect that many rotations with pivot positions far from diagonal will be skipped. In fact, in each row of the row-cyclic pivoting, each skipped rotation increases the probability that the next rotation in that row will be skipped as well. Hence, it is reasonable to have a pivot strategy which will dynamically adapt to the structure of the matrix and anticipate small pivots. This strategy saves many unnecessary dot products, and it can be applied inside the blocks of the quasi-cyclic strategy (2.5) as well as globally on $X$.

The scheme on Figure 2.2 gives the main idea. The basic strategy is row-cyclic with de Rijk's pivoting[1] [9], but if in a row $i$ a certain number of consecutive rotations is skipped (because the pivot elements are sufficiently small), the control of the row-cycling moves to the next row—the remaining pivot positions of the $i$th row are not visited. This strategy is motivated by the following reasons. First, it is very likely that the $\otimes$-positions in Figure 2.2 will pass the tolerance check, so we save unnecessary dot products. Second, even if the $\otimes$-positions do not satisfy the tolerance criterion, they are expected to be much smaller than the pivot positions closer to the diagonal, and it is more useful for the overall convergence to reduce those positions close to the diagonal.

---

[1] Due to preconditioning and monotonicity of Jacobi rotation (it increases larger and decreases smaller diagonal pivots), de Rijk's pivoting is identity most of the time.

Certainly, this modification of the row-cyclic strategy may bring no savings in the pathological case. However, it does no harm—in that case it simply reduces to the classical strategy. Also, this modification is in general not convergent, so an additional switch turns it off after at most 3 or 4 modified sweeps. After that, the classical full sweep does the final cleanup. Numerical evidence shows that in a nonpathological case the expected total number of rotations in 3 or 4 sweeps of this predict-and-skip strategy is much smaller than in one classical full sweep.

**2.5. Cache-aware pivot strategy.** In sections 2.1–2.4 we considered modifications to reduce the number of operations and to improve the convergence rate of the Jacobi SVD algorithm. Numerical evidence shows that those modifications, combined with the preconditioning, substantially improve the one-sided Jacobi SVD algorithm. However, the algorithm still transforms a full square array, where the basic operations (dot product and plane rotation) both have $O(n)$ operations per $O(n)$ memory references. Pivot strategies in Jacobi methods are usually not designed to enhance temporal and spatial data locality, which results in numerous cache misses, thus degrading the performance. Fortunately, a well-known tiling technique fits very nicely into our previous modifications and considerably improves data locality.

Introduce a parameter $b$ (block size expressed in number of columns) and partition the columns of $X$ in $\lceil n/b \rceil$ blocks (the first $\lceil n/b \rceil - 1$ blocks with $b$ columns each, the last block with the remaining $n - b(\lceil n/b \rceil - 1)$ columns). This introduces a $\lceil n/b \rceil \times \lceil n/b \rceil$ block partition in $H = X^T X$ and the new strategy is to visit all blocks in the usual row-cyclic fashion (on the block-level), where at the beginning of the $r$th block row, after rotating in the diagonal block $(r, r)$, we allow the possibility of transforming the next $k$ diagonal blocks (and, optionally, to repeat transformations in the block $(r, r)$) before entering the block $(r, r + 1)$. The parameter $k$ is a small integer (typically $0, 1, 2$) depending on $X$, $n$, $b$ and cache parameters. It influences the convergence rate (this is easily seen by taking, for example, $k = 1$) and memory access patterns. Inside each block all positions are visited row by row. We call this strategy "tiled row-cyclic." Its detailed description is shown in Algorithm 1.

Proposition 2.4. *The tiled row-cyclic strategy with $k = 0$ is equivalent to the row-cyclic strategy: both strategies compute the same matrix after the full cycle of $n(n-1)/2$ rotations. Thus, it is convergent.*

The proof is straightforward. The proof of global convergence for $k > 0$ is only a technical matter [22]. The tiled row-cycling can be immediately deployed inside the blocks of (2.5), and it can be modified following the lines of section 2.4.

## 3. Underflow and overflow—problems and solutions by perturbation theory.

Reliable software implementation of an SVD algorithm must take care of underflow and overflow exceptions. This is particularly important for our new preconditioned Jacobi SVD algorithm (cf. [14, Algorithm 4] with the triangular Jacobi SVD as described in previous sections), because it is designed to compute the singular values in the full range of floating point numbers. For example, if $\sigma_{\max}(A) \approx 10^{30}$, $\sigma_{\min}(A) \approx 10^{-30}$, but $\min_{D=\mathrm{diag}} \kappa_2(AD)$ is moderate, then we can approximate all singular values to high relative accuracy even in single precision arithmetic (with round-off unit $\varepsilon \approx 10^{-8}$). (For comparison, bidiagonalization-based methods cannot guarantee any correct digit in the singular values below $\varepsilon\sigma_{\max}$, which is in this case approximately $10^{22}$.) Jacobi SVD computation in the full range of floating point numbers requires nonstandard implementation of Jacobi rotation because it can get denormalized or flushed to identity, even in cases where its action is nontrivial [11].

In this section we discuss some other problems related to underflow and overflow.

**Algorithm 1** Tiled row-cyclic strategy with tile size $b$.

```
{Simplified description of one full sweep}
$N_{bl} = \lceil n/b \rceil$
for $r = 1$ to $N_{bl}$ do
  $i = (r-1) \cdot b + 1$
  for $d = 0$ to $k$ do {Do the blocks $(r,r), \ldots, (r+k, r+k)$}
    $i = i + d \cdot b$
    for $p = i$ to $\min\{i+b-1, n\}$ do
      for $q = p+1$ to $\min\{i+b-1, n\}$ do
        rotate pivot pair $(p,q)$
      end for
    end for
  end for
  $i = (r-1) \cdot b + 1$
  for $c = r+1$ to $N_{bl}$ do
    $j = (c-1) \cdot b + 1$
    for $p = i$ to $\min\{i+b-1, n\}$ do
      for $q = j$ to $\min\{j+b-1, n\}$ do
        rotate pivot pair $(p,q)$
      end for
    end for
  end for
end for
```

The underflow and overflow thresholds are denoted by $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$, respectively. The round-off unit of the working precision is denoted by $\boldsymbol{\varepsilon}$.

**3.1. Scaling against overflow.** Overflow issues are resolved simply by multiplying the matrix by a suitable scalar factor. However, even this simple operation can introduce unacceptably large errors. For instance, LAPACK's driver routine xGESVD computes $\alpha = \max_{i,j} |A_{ij}|$ and scales the input matrix $A$ with $(1/\alpha)\sqrt{\boldsymbol{\nu}}/\boldsymbol{\varepsilon}$ (if $\alpha < \sqrt{\boldsymbol{\nu}}/\boldsymbol{\varepsilon}$) or with $(1/\alpha)\boldsymbol{\varepsilon}\sqrt{\boldsymbol{\omega}}$ (if $\alpha > \boldsymbol{\varepsilon}\sqrt{\boldsymbol{\omega}}$).

*Example* 3.1. Take in MATLAB $A = \left(\begin{smallmatrix} 1.0e250 & 0 \\ 0 & 1.0e{-201} \end{smallmatrix}\right)$, $d = \mathrm{diag}(A)$, $\sigma = \mathrm{svd}(A)$. $A$ is (bi)diagonal, and its singular values are on the diagonal. However,

$$d = \begin{pmatrix} 9.999999999999999e+249 \\ 1.000000000000000e-201 \end{pmatrix}, \quad \sigma = \begin{pmatrix} 9.999999999999999e+249 \\ 1.0000000000\underline{16167}e-201 \end{pmatrix}.$$

To explain this, let $\alpha = \max_{i,j} |A_{ij}|$, $\boldsymbol{\varepsilon} = eps/2$, $\boldsymbol{\omega} = realmax$, $\boldsymbol{\nu} = realmin$, $s = \boldsymbol{\varepsilon}\sqrt{\boldsymbol{\omega}}/\alpha$, and scale $A$ with $s$. The singular values of $sA$ are on its diagonal; scaling the diagonal of $sA$ with $1/s$ changes the $(2,2)$ entry precisely to $1.000000000016167e$–$201$. Five digits in the second singular value of a $2 \times 2$ diagonal matrix are lost due to scaling $\sigma = (1/s) * (s * d)$. (In MATLAB, $\boldsymbol{\omega} \approx 1.79 \cdot 10^{308}$, $\boldsymbol{\nu} \approx 2.22 \cdot 10^{-308}$.) The problem is not removed if $s$ is changed to the closest integer power of 2. Note that in this example $\lambda = \mathrm{eig}(A)$ returns $\lambda = d$.

Our implementation of fast scaled Jacobi rotations uses the column norms and the cosines of the angles between the columns (cf. [11]); i.e., we can compute the singular values of $X$ even if $\sigma_{\min}(X) \approx \boldsymbol{\nu}$, $\sigma_{\max}(X) \approx \boldsymbol{\omega}$. Since the largest singular value of an $n \times n$ $X$ is bounded by $\sqrt{n} \max_j \|X(:,j)\|_2$, it is enough to have initial $X$ scaled so that its maximal column is not larger than $\boldsymbol{\omega}/\sqrt{n}$. Since the largest value of any column norm of $X$ is $\sqrt{n}\boldsymbol{\omega}$, even in the most extreme case, the scaling factor against overflow does not have to be smaller than $1/n$. In the nonextreme case $(\max_j \|X(:,j)\|_2/\min_j \|X(:,j)\|_2 < \boldsymbol{\omega}\boldsymbol{\varepsilon}/\sqrt{n})$ we scale $X$ to have maximal column norm at $\sqrt{\boldsymbol{\omega}}/\sqrt{n}$.

*Remark* 3.1. A concrete implementation of BLAS and LAPACK may contain computational routines which are optimized for speed at the cost of reduced computational range. For instance, xNRM2() is sometimes implemented as SQRT(xDOT()), which works correctly for vector norms in the range $(\sqrt{\boldsymbol{\nu}}, \sqrt{\boldsymbol{\omega}})$. In such cases, scaling of $A$ must ensure that the maximal column is not larger than $\sqrt{\boldsymbol{\omega}}/\sqrt{n}$ in the Euclidean norm. If the spectrum of singular values spreads over the full range of normalized numbers and if all of them are wanted to high relative accuracy, then enforcing $\sqrt{\boldsymbol{\omega}}/\sqrt{n}$ as the maximum column norm may damage smallest singular values.

*Remark* 3.2. The one-sided Jacobi SVD can be adapted to work even beyond the range of working precision. Let $A = A_0 D_0$ with diagonal $D_0$, where $A$ cannot be stored because of underflow/overflow, but both $A_0$ and $D_0$ can. Fast scaled Jacobi rotations work on the pair $A_0, D_0$ and deliver the result in factored form. The only required modification is that the array of scaling factors for fast rotations be initialized to $\mathrm{diag}(D_0)$ instead of a vector of ones.

**3.2. An unusual underflow problem.** Preliminary tests of software implementation of the preconditioned Jacobi SVD algorithm showed undesirable behavior in some cases of strongly graded matrices: the convergence of the Jacobi iterations was swift, the total number of rotations was very small, but the run time was unacceptably long. This called for detailed step-by-step analysis. Recall that the algorithm first computes[2] $AP = Q_0\binom{R}{0}$, then it computes the QR factorization $L = QT$ of $L = R^T$, and finally it applies our new Jacobi SVD algorithm to the lower triangular matrix $X = T^T$.

To our surprise, in some examples the second QR factorization without pivoting was considerably slower than the QR factorization with column pivoting. In these examples, $L$ was structured, strongly graded, with deep gap in the spectrum:

$$L = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix}, \quad \text{with } \|L_{11}^{-1}\|_2^{-1} \gg \|L_{22}\|_2 \text{ and } \|L_{21}\|_2 \ll \|L_{11}\|_2.$$

We traced the problem to computation with many denormalized floating point numbers during the computation of the factorization $L = QT$. From the block-partitioned QR factorization $L = \left(\begin{smallmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{smallmatrix}\right)\left(\begin{smallmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{smallmatrix}\right)$ we have $T_{12} = Q_{11}^T L_{11}^{-T} L_{21}^T L_{22}$ and $\|T_{12}\|_2 \leq \|Q_{11}\|_2 \|L_{11}^{-1}\|_2 \|L_{22}\|_2 \|L_{21}\|_2$. This is the well-known QR mechanism that eventually forces the $(1,2)$ block to zero, and it is clearly seen in Example 3.2.

*Example* 3.2. This example illustrates how denormalized numbers appear in the QR factorization. Using SGEQRF from LAPACK we compute

$$L = \begin{pmatrix} 1.0e{+}20 & 0 & 0 \\ 1.0e{-}15 & 1.0e{-}06 & 0 \\ 1.0e{-}20 & 1.0e{-}25 & 1.0e{-}21 \end{pmatrix}, \quad T = \begin{pmatrix} -0.10e{+}21 & -0.99e{-}41 & 0.00e{+}00 \\ 0 & -0.10e{-}05 & -0.99e{-}40 \\ 0 & 0 & 0.99e{-}21 \end{pmatrix}.$$

From the point of view of numerical accuracy, these denormalized numbers in the upper triangle of $T$ are as good as zeros—backward stability and the forward errors are given with respect to column norms, and these are well preserved if the entries

---

[2] For simplicity, we give only one branch of the algorithm.

of the initial $A$ are normalized numbers. Unfortunately, if the denormals are created and transformed inside an optimized black-box routine, they can cause considerable slowdown (take, e.g., $n > 1000$).

The performance of Jacobi rotations applied to a lower triangular matrix can also be degraded by the denormalized numbers. The dot products needed to compute the rotation angles can be extremely slow (many of the summands can be denormalized and are as good as zeros in the final result), and rotations with small angles during the first greedy sweep may generate additional denormalized entries where there are zeros in the upper triangle.

**3.2.1. Solution by artificial perturbation.** How do we deal with this problem? We can ignore it, because it is automatically solved with proper implementation of the machine arithmetic or with the *set to zero* underflow. However, it is very instructive to solve it in the framework of the imperfect arithmetic. The first natural idea is to set small off-diagonal matrix entries to zero. This has to be done by inspection after each transformation in QR factorization (or in the Jacobi iterations). This may kill the performance of highly optimized blocked code, and the underflows may still appear because they actually grow in places of zero entries. Furthermore, setting an entry to zero means perturbing the data, which may not be allowed because it causes unacceptably large perturbation. (For instance setting in Example 3.2 the element $L_{31}$ to zero introduces large perturbation in the third row of $L$. In some cases this may be unacceptable.)

Now consider the opposite approach—using artificial perturbation we destroy all zeros and increase small entries. Let $X$ be an $n \times n$ lower triangular and nonsingular matrix. The goal is to replace $X$ with $X + \delta X$, where the perturbation $\delta X$ is (i) small enough so that it does not introduce errors larger than the initial uncertainty of the SVD caused by computing $X$; (ii) big enough to prevent underflows in the next QR factorization or in Jacobi iterations; (iii) small enough so that it does not interfere with the convergence and that it does not prevent the use of the lower triangular structure; and (iv) small enough so that it does not preclude stable a posteriori computation of the right singular vectors.

This means that the perturbation $\delta X$ has to be columnwise and also rowwise small. We use $\zeta$ to denote an appropriate threshold value used in the construction of $\delta X \equiv \delta X_\zeta$, for instance, $\zeta = \sqrt{\boldsymbol{\nu}}$, $\zeta = \sqrt{\boldsymbol{\nu}/\boldsymbol{\varepsilon}}$, or $\zeta = \boldsymbol{\varepsilon}/n$. The choice of $\zeta$ depends on $\kappa_2(X)$, from smallest values in the nonextreme case, $\kappa_2(X) < \boldsymbol{\omega\varepsilon}$, to largest in the case[3] $\kappa(X) \approx \omega^2\boldsymbol{\varepsilon}$.

Definition 3.1. *For lower triangular matrix $X$ and small positive parameter $\zeta$, define $\delta X_\zeta$ as follows:*

$$(\delta X_\zeta)_{ij} = \left\{ \begin{array}{cc} 0 & \text{if } |X_{ij}| \geq \zeta \min\{|X_{ii}|, |X_{jj}|\} \\ -X_{ij} + \operatorname{sign}(X_{ij})\zeta \min\{|X_{ii}|, |X_{jj}|\} & \text{else} \end{array} \right\} \text{ for } i \geq j; \tag{3.1}$$

$$(\delta X_\zeta)_{ij} = -\operatorname{sign}(X_{ji})\zeta \min\{|X_{ii}|, |X_{jj}|\} \text{ for } i < j. \tag{3.2}$$

The perturbed matrix $\tilde{X} = X + \delta X_\zeta$ is not triangular. On the other hand, efficient implementation of the first sweep of the quasi-cyclic strategy (section 2.2) is based on the triangular form. We now show that in application of the pivoting (2.5) to $\tilde{X}$ we can use the same technique as in section 2.2 and treat $\tilde{X}$ as triangular. More

---

[3] Extreme cases are relevant only if the singular values are well determined by the data and all wanted to high relative accuracy. Of course, such extreme cases are difficult if the machine arithmetic is not well implemented.

precisely, in the first bulk of rotations $\tilde{X}_{[3]}^{\circlearrowleft}$ we transform only $\tilde{X}_{[3]}(n_1+n_2+1:n,:)$, thus ignoring the perturbation added to $X_{[3]}(1:n_1+n_2,:)$. The perturbed zeros inside the block $\tilde{X}_{[3]}(1:n_1+n_2,:)$ are used to drown denormalized numbers during the phase $X^{[1]} \leftrightarrow X^{[2]}$. The same strategy is applied to $\tilde{X}_{[4]}^{\circlearrowleft}$, where in particular $\tilde{X}_{[4]}(1:n_1+n_2+n_3, 1:n_4)$ is treated as zero and not transformed. In other words, we ignore the perturbation $\delta X_\zeta$ whenever we need a triangular structure to apply (2.5) as described in section 2.2.

Proposition 3.2. *The application of the first sweep of the quasi-cyclic strategy* (2.5) *to* $\tilde{X}$ *(as described above) is rowwise backward stable. Further, the perturbation of the upper triangle can be ignored in the a posteriori computation of the right singular vectors.*

*Proof.* To justify this manipulation with $\delta X_\zeta$, we first note that it contains tiny rowwise relative perturbations $X$. Indeed, for $i<j$ we have

$$\frac{|(\delta X_\zeta)_{ij}|}{\|X(i,:)\|_2} = \zeta \frac{\min\{|X_{ii}|,|X_{jj}|\}}{\|X(i,:)\|_2} \leq \zeta, \text{ where in case of a graded matrix } \frac{|X_{jj}|}{\|X(i,:)\|_2} \ll 1.$$

The key argument is shown for the simplest $(n_1,n_2)$ block partition of $\tilde{X}$,

$$\tilde{X} = \begin{pmatrix} \tilde{X}_{11} & \tilde{X}_{12} \\ \tilde{X}_{21} & \tilde{X}_{22} \end{pmatrix} = \begin{pmatrix} \tilde{X}_{11} & 0 \\ \tilde{X}_{21} & \tilde{X}_{22} \end{pmatrix} + \begin{pmatrix} 0 & \tilde{X}_{12} \\ 0 & 0 \end{pmatrix} \equiv \tilde{X}_{\text{L}} + \delta\tilde{X}_{\text{L}}.$$

Obviously, $\|\tilde{X}_{12}(i,:)\|_2 \leq \sqrt{n_2}\zeta\|X(i,:)\|_2$ for all $i=1,\ldots,n_1$. Suppose we transform the last $n_2$ columns of $\tilde{X}$, following some pivot strategy, but we simply do not reference $\tilde{X}_{12}$ (that is, we work on $\tilde{X}_{\text{L}}$ and use its triangular structure). The computed matrix is represented by the backward perturbation analysis as

$$\begin{aligned} \tilde{X}' &= \begin{pmatrix} \tilde{X}_{11} & 0 \\ \tilde{X}_{21} & \tilde{X}_{22}+\delta\tilde{X}_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \hat{W} \end{pmatrix} + \begin{pmatrix} 0 & \tilde{X}_{12} \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} \tilde{X}_{11} & \tilde{X}_{12}\hat{W}^T \\ \tilde{X}_{21} & \tilde{X}_{22}+\delta\tilde{X}_{22} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \hat{W} \end{pmatrix}, \quad \hat{W}^T\hat{W} = I, \end{aligned}$$

where $\|\delta\tilde{X}_{22}(i,:)\|_2 \leq O(n)\varepsilon\|\tilde{X}_{22}(i,:)\|_2$, $i=1,\ldots,n_2$, which means that ignoring $\tilde{X}_{12}$ is equivalent to replacing it with $\tilde{X}_{12}\hat{W}^T$ and then applying $\hat{W}$. Since right-handed orthogonal transformation does not change the row-norms of the involved matrices, the rowwise backward stability is preserved. ☐

If we need only to compute the QR factorization of $X$, or to compute only the singular values, then we can allow even bigger $\delta X_\zeta$. Each $X_{ij}$ in the lower triangle with the property $|X_{ij}| < \zeta|X_{jj}|$ is replaced with $\text{sign}(X_{ij})\zeta|X_{jj}|$, and thus $(\delta X_\zeta)_{ij} = -X_{ij} + \text{sign}(X_{ij})\zeta|X_{jj}|$ for all $i>j$. Simultaneously, the position $X_{ji}$ in the upper triangle is set to $(\delta X_\zeta)_{ji} = -\text{sign}(X_{ij})\zeta|X_{ii}|$. Note that $\|\delta X_\zeta(:,j)\|_2 \leq \sqrt{n}\zeta|X_{jj}|$, i.e., the perturbation is columnwise small. Since the matrix $X$ is computed in the QR factorization with pivoting, the condition number of column scaled $X$ is moderate, which means that computations with $X$ and $X+\delta X_\zeta$ will give equally good singular value approximations.

## 4. Numerical testing.

In this section we present software implementation of the new algorithm described in [14, Algorithm 4] and in this report. We give the results of preliminary testing of the algorithm with respect to numerical accuracy and efficiency (run times compared with those of existing algorithms). Our goal is numerically reliable software implemented to reach a reasonable fraction of the efficiency of the less accurate bidiagonalization-based methods. In other words, we want to make the high accuracy of the Jacobi SVD algorithm so affordable that the new algorithm becomes attractive as one of the methods of choice for dense full SVD computation.

Carefully designed testing of the software is also a test of the theory. It shows how sharp the theoretical bounds are and also gives new insights into the cases of input matrices which are on the boundaries of the theoretical assumptions. Good test cases give insights into the behavior of the algorithm and may induce modifications which improve the efficiency of the algorithm. The feedback loop created in this way is part of the research process. In fact, the material of section 3 is the result of numerical tests of an early version of the code. Further, during the tests of our code we found serious problem in the LAPACK implementations xGEQPF and xGEQP3 of the QR factorization with column pivoting; see [13] for a provably stable implementation.

We test single precision (32-bit representation, $\varepsilon \approx 5.3 \cdot 10^{-8}$) implementation. It is always assumed that the nonzero entries of input matrix are normalized floating point numbers.

**4.1. Measuring error—distance to what?** One difficulty in testing a new SVD software on a large set of pseudorandom matrices is how to provide reference (exact) values of $\Sigma$, $U$, and $V$, which are used to estimate the accuracy of the computed approximations $\tilde{\Sigma}$, $\tilde{U}$, $\tilde{V}$. One could start by generating pseudorandom numerically orthogonal $\overline{U}$, diagonal $\overline{\Sigma}$, and numerically orthogonal $\overline{V}$ and then define $A = \mathit{computed}(\overline{U}\ \overline{\Sigma}\ \overline{V}^T)$. However, the numerical SVD of $A$ may be very much different from $\overline{U}\ \overline{\Sigma}\ \overline{V}^T$. Using the same algorithm in higher precision is useful but not always—depending on the matrix, it is possible that both procedures compute with large errors. The alternative is to use existing, tested(!), and trusted double precision software to compute the SVD of a given test matrix $A$. In our case, this means DGESVD and/or DGESDD[4] from LAPACK, but this will be useful only as long these procedures guarantee at least eight digits of accuracy, that is, for (roughly) $\kappa(A) < 1/\varepsilon \approx 10^8$. Our choice of the reference procedure is the classical one-sided Jacobi SVD with de Rijk's [9] pivoting, implemented in double precision.

If $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$ and $\hat{\sigma}_1 \geq \cdots \geq \hat{\sigma}_n$ are the computed and the reference singular values computed in higher precision, then the forward errors of interest are[5]

$$\mathbf{e}_i = \frac{|\tilde{\sigma}_i - \hat{\sigma}_i|}{\hat{\sigma}_i}, \quad i = 1, \ldots, n, \quad \mathbf{e} = \max_{i=1:n} \mathbf{e}_i. \tag{4.1}$$

**4.2. Test matrices.** Our primary targets are the matrices of the form $A = BD$, where $D$ is diagonal and $B$ is well conditioned with equilibrated (unit in Euclidean norm) columns. In that case the relative error in the output is governed by the condition number $\kappa(B)$ independent of $D$. To illustrate this property we need to generate test matrices $A = BD$, where $B$ has given $\kappa(B)$ and unit columns. Moreover, the matrices should be generated so systematically that the maximal measured forward errors attain the predicted theoretical bounds, and that experimental data show that no accuracy can be guaranteed if the assumptions of the theory are not satisfied. In

---

[4] During the testing we accidentally found an example of serious failure of the DGESDD procedure from the SUN performance library—a ghost singular value of the size of the largest one appeared in the dominant part of the spectrum.

[5] Here by definition $0/0 = 0$.

that case we will have experimental evidence that both the theory and the numerical testing are done properly.

We use the algorithm of Stewart [29] to generate random orthogonal matrices distributed uniformly with respect to the Haar measure over the orthogonal group $\mathcal{O}(n)$. If $W_1$ and $W_2$ are two such matrices, and if $S$ is diagonal with given condition number $\kappa(S)$, we compute $C = W_1 S W_2$. Then we use the fact that for the matrix $C^T C$ there always exists an orthogonal $W_3$ such that the diagonal entries of $W_3^T(C^T C)W_3$ are all equal to $\mathrm{Trace}(C^T C)/n$. Then the matrix $B = CW_3$ has equilibrated columns and condition number $\kappa$. If we generate diagonal $D$, then $A = BD$. There are several ways to generate the matrix $W_3$; see, e.g., [5], [7]. The distributions of the diagonal entries of $S$ and $D$ can be chosen in different ways. We use the modes provided in the LAPACK test matrix generators (parameter MODE in DLATM1), and the chosen modes are denoted by $\mu(S)$ and $\mu(D)$. Thus, each generated matrix $A$ has four parameters, $p(A) = (\kappa(S), \mu(S), \kappa(D), \mu(D))$. For each fixed $p(A)$, we use three different random number generators provided in the LAPACK testing library (LAPACK/TESTING/MATGEN/), and with each of them we generate a certain number of samples (test matrices). In this way, we have an automated generator of pseudorandom matrices with certain relevant parameters varying systematically in a given range; for instance, $\kappa(S)(= \kappa(B))$ is set to take the values 10, $10^2$, $10^3, \ldots, 10^8$.

**4.3. Test results.** Our new algorithm is implemented in a LAPACK-style routine SGEPVD, which is in an early stage of development. We have done no serious profiling in order to optimize it for a particular architecture. The bulk of the code (rotations) is still on BLAS 1 level, and we have plans to change this in near future. Nevertheless, the obtained results are surprisingly good and encouraging. The results presented in this report were obtained on an HP X2100 workstation (1.9GHz Pentium 4, 1Gb RAM, 8 Kb L1 cache, 256 Kb L2 cache), using GNU FORTRAN compiler F77 3.3.1 under SuSE Linux. The LAPACK library is compiled from the source code and linked with the BLAS library from Intel's MKL 6.1.

**4.3.1. Computing the full SVD.** The test matrices of the form $A = BD$ are generated as follows. We take $A = ((W_1 S W_2) W_3) D$ as described in section 4.2, and $\kappa(S) = 10^i$, $i = 1, \ldots, 8$, and $\kappa(D) = 10^{2j}$, $j = 0, \ldots, 7$. For each fixed pair $(i, j)$ we generate diagonal $S$ and $D$ each with four different distributions of the diagonal entries (as specified by the parameter MODE). This gives for each fixed $(\kappa(S) = 10^i, \kappa(D) = 10^{2j})$ 16 different types of matrices, giving a total of $64 \cdot 16 = 1024$ classes. The matrices are generated in four nested loops; the outer loop controls $\kappa(B) = \kappa(S)$. Hence, the matrices are divided into eight groups with fixed $\kappa(B)$.[6] Finally, we choose the row and the column dimensions, $m$ and $n$, and the test procedure is ready.

Once we compute $A \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T$, $\tilde{\Sigma} = \mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n)$, we can immediately estimate the quality of the computed decomposition using the following computed quantities:

$$\mathbf{r} = \mathit{computed}\left(\frac{\|A - \tilde{U}\tilde{\Sigma}\tilde{V}^T\|_F}{\|A\|_F}\right) \quad \text{(should be at most } f(m,n)\boldsymbol{\varepsilon},\ f \text{ moderate);}$$

$$\mathbf{o}_U = \mathit{computed}\left(\max_{i,j} |(\tilde{U}^T\tilde{U})_{ij} - \delta_{ij}|\right) \quad \text{(should be at most } O(m\boldsymbol{\varepsilon})\text{);}$$

$$\mathbf{o}_V = \mathit{computed}\left(\max_{i,j} |(\tilde{V}^T\tilde{V})_{ij} - \delta_{ij}|\right) \quad \text{(should be at most } O(n\boldsymbol{\varepsilon})\text{).}$$

---

[6] This helps in the interpretation of the results of the experiments and explains the shapes of the graphs on the figures given here.

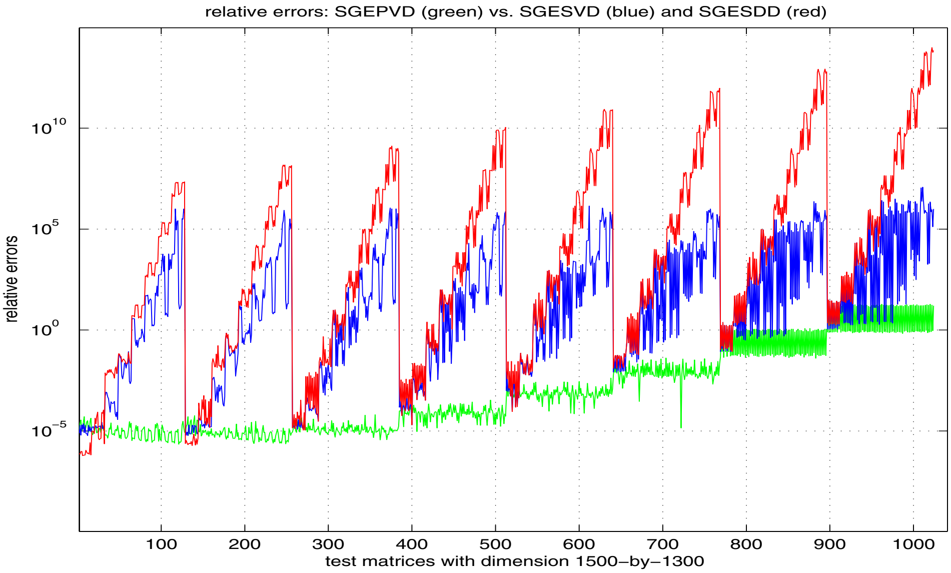


FIG. 4.1. *Maximal relative errors in the computed singular values for* $1500 \times 1300$ *matrices. The top curve (worst case) describes the accuracy of SGESDD. The middle curve represents the errors of SGESVD, and the lowest curve (smallest relative errors) belongs to SGEPVD.*

These measures are useful to test the correctness of the code and backward stability in the matrix norm sense. It is easy to show that $\mathbf{r}$, $\mathbf{o}_U$, and $\mathbf{o}_V$ can be computed sufficiently accurately (best using higher precision) to be used as relevant measures of the quality of the computed decomposition. Thus, the standard error bound can be a posteriori numerically checked. Our procedure, called SGEPVD,[7] has successfully passed all three tests; $\mathbf{r}$, $\mathbf{o}_U$, $\mathbf{o}_V$ were in the allowed ranges. Note that SGEPVD returns the singular vectors numerically orthogonal up to the theoretical bound $m\varepsilon$ ($n\varepsilon$), which, as in other algorithms dealing with numerical orthogonality, for large $m$ and $n$ may not look satisfactory in single precision ($\varepsilon \approx 10^{-7}$ and, e.g., $m = n = 4000$).

Before we go over to the comparison of SGEPVD with SGESVD and SGESDD from LAPACK, we should point out that SGEPVD computes the SVD to higher accuracy and also provides an estimate of the maximal relative error by computing an approximation of $\|B^\dagger\|_2$.

We show only two out of the many tests performed during code development. Or first test ran with $m = 1500$, $n = 1300$, and with one pseudorandom matrix in each class. Compare the maximal relative errors in computed singular values for all 1024 test cases, shown on Figure 4.1. It is clearly seen that the accuracy of SGEPVD depends on $\kappa(B)$, while the other two methods depend on $\kappa(A)$. Any SVD algorithm that starts with bidiagonalization is at risk to have error behaving like the SGESVD and SGESDD errors in Figure 4.1. The best caption for this figure is the title of [8]. Note that SGESVD returns much better results than SGESDD. To the best of our knowledge, this fact is not mentioned elsewhere in the literature. Also, note the considerable upward bias in the relative errors of the bidiagonalization-based procedures (cf. [30]).

The timings for this example are shown on Figure 4.2. We immediately note that the new Jacobi SVD algorithm is not that much slower than the bidiagonalization-

---

[7] PVD is the acronym for principal value decomposition, an old name for SVD.

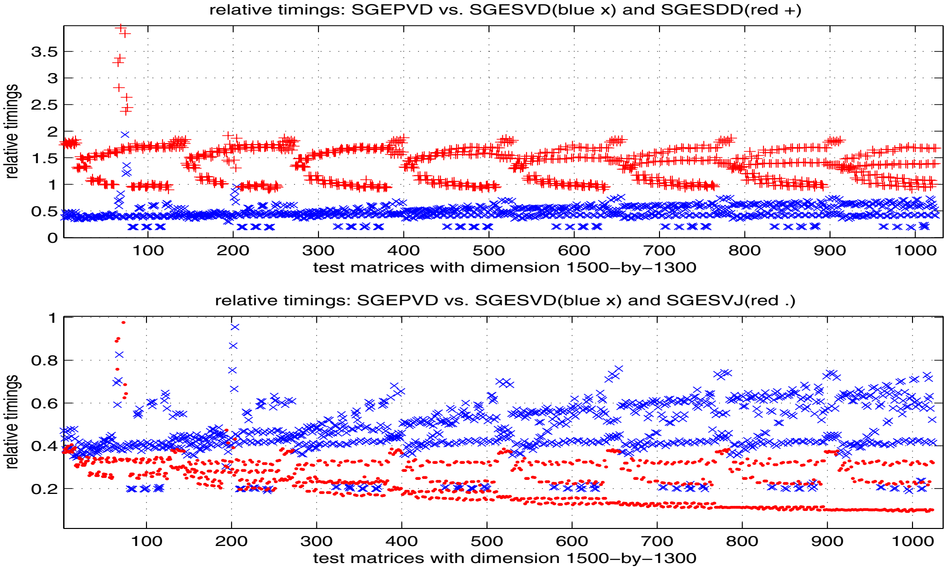


FIG. 4.2. *Computing full SVD: Relative timings for* $1500 \times 1300$ *matrices on a Pentium* 4 *machine with Intel MKL* 6.1 *library. In the first plot, the crosses denote* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVD}}$ *and the pluses* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESDD}}$. *The second plot shows* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVD}}$ *(crosses) and* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVJ}}$ *(dots). See Remark* 4.1.

based fast methods. In fact, it outperforms the QR algorithm and is on average less than twice as slow as the divide-and-conquer algorithm. The worst-case performance for SGEPVD is on matrices with weak column scaling and a singular spectrum composed of many tight clusters (examples above the 1.5 mark on Figure 4.2). In all other cases the time of SGEPVD is on average 1.5 times the time of SGESDD. Here, again, we stress the fact that the results obtained by SGEPVD enjoy much better numerical properties and that the time of SGEPVD includes computed error bound—better results and additional information are computed in reasonable time. Thus, for a fair comparison one should consider both Figures 4.1 and 4.2 before deciding which algorithm is the better choice for a particular application. The second plot on Figure 4.2 shows $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVD}}$ and $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVJ}}$, where SGESVJ denotes our careful implementation (cf. [11]) of de Rijk's [9] one-sided Jacobi SVD. (De Rijk's one-sided Jacobi SVD is much more efficient than other classical cyclic Jacobi algorithms.) It is interesting that de Rijk reported superior performance of his Jacobi SVD code over the SSVDC routine from LINPACK (Golub–Reinsch SVD) on CYBER 205. Note that our code SGEPVD runs up to ten times faster than SGESVJ.

In the second test we have $500 \times 350$ matrices, with two examples in each of 1024 classes. The results are given in Figure 4.3.

*Remark* 4.1. Note the few outliers above mark 2 on Figures 4.2 and 4.3. They correspond to matrices on which SGEPVD actually performed very well, with a low number of rotations and swift convergence. (We checked this by inspecting the details of those particular runs. In fact, on these matrices even the classical one-sided Jacobi, usually much slower, comes close to our new method.) However, since our threshold for perturbation used to trap denormals was set to $\zeta = \sqrt{\boldsymbol{\nu}} \approx 10^{-19}$, some of them were

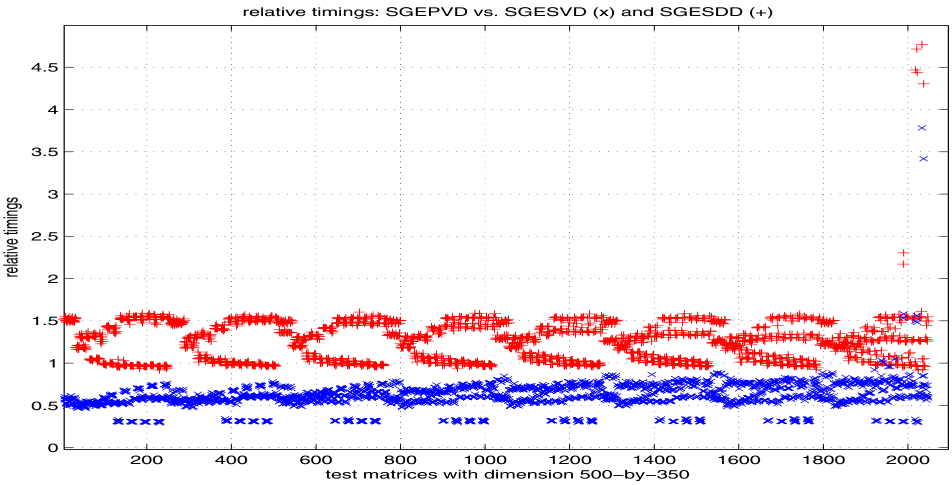


FIG. 4.3. *Computing full SVD: Relative timings for* $500 \times 350$ *matrices on a Pentium* 4 *machine with Intel MKL* 6.1 *library. The crosses denote* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVD}}$ *and the pluses are* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESDD}}$.

not captured, and imperfect denormalized arithmetic caused considerable slowdown. We used this value of $\zeta$ to illustrate the problem. In practice the threshold can be set higher; e.g., if in those cases $\zeta = \varepsilon/n$, or if *set to zero* underflow is in function, then the run time reflects the actual flop count and the outliers disappear.

*Remark* 4.2. We have noted that using a double accumulated dot product in preparation of Jacobi rotation reduces the total number of rotations, especially in cases of multiple or tightly clustered singular values. Unfortunately, in the MKL BLAS library DSDOT performs poorly in comparison to SDOT, and the savings in number of rotations do not reduce the total run time.

*Remark* 4.3. The results may vary on the same machine with different BLAS libraries. We note that with the GOTO BLAS [18], [19] all routines run faster (as compared to the MKL library), but the relative speedup is smallest in our algorithm. This is because GOTO BLAS 3 outperforms MKL 6.1 BLAS 3, but GOTO BLAS 1 is no match for really efficient MKL 6.1 BLAS 1. Since our code still depends on BLAS 1 (dot products and plane rotations), switching to GOTO BLAS has mixed consequences. A hybrid of the two libraries would be much better for our algorithm.

**4.3.2. Computing only Σ.** We have established that the singular values are computed by the new algorithm as predicted by the theory—our variant of the Jacobi SVD complies with [8], [10]. In the previous section we showed that it computes the full SVD with efficiency comparable to fast bidiagonalization-based approaches. If only the singular values are needed, reaching a similar level of relative efficiency seems to be mission impossible, for we would need a Jacobi-based algorithm that computes $\Sigma$ in time comparable to the time needed to bidiagonalize the matrix! However, the results of [14, Algorithm 1] with the rank revealing QR factorization [4], [3] are encouraging.

The test matrices of the form $A = BD$ are generated as in section 4.3.1, with 2048 examples in each test run. We show the results of two tests; in the first the matrices were $500 \times 350$ and in the second $m = 1000$, $n = 700$. The maximal measured relative errors $\mathbf{e}$ are not shown because they behave as shown in section 4.3.1. In Figure 4.4

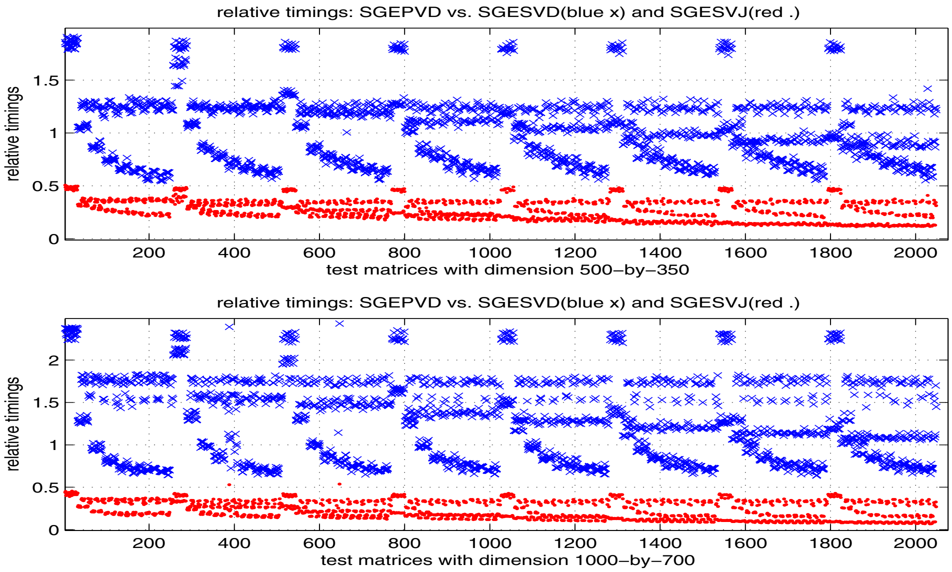


FIG. 4.4. *Computing only* $\Sigma$*: Relative timings for* $500 \times 350$ *and* $1000 \times 700$ *matrices on a Pentium* 4 *machine with Intel MKL* 6.1 *library. The crosses denote* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVD}}$ *and the dots are* $time_{\mathrm{SGEPVD}}/time_{\mathrm{SGESVJ}}$.

we display the relative timings and compare SGEPVD with SGESDD (or SGESVD) and with the classical one-sided Jacobi SVD with de Rijk's pivoting (SGESVJ). The speedup of the new algorithm over SGESVJ ranges from a factor of 2 up to a factor of 15. (We note that SGEPVD computes $\Sigma$ and also an estimate of the scaled condition number.) It is interesting to note that in cases of well separated singular values (especially in cases of badly scaled matrices) the new method computes all singular values faster than bidiagonalization-based methods. In less favorable cases of clustered singular values the new method is slower with a factor of up to 2.43, depending on the input matrix.

## 5. Conclusion and future work.

The most important message of [14] and this paper is that the question of the ultimate dense nonstructured full SVD method is still open. What is desired is the most efficient algorithm capable of computing the SVD to optimal (numerically feasible) accuracy warranted by the data. Because the one-sided Jacobi SVD is more accurate than any other algorithm that first bidiagonalizes the matrix, in our quest for the ultimate dense nonstructured SVD algorithm we follow the Jacobi idea. We have enhanced the basic Jacobi SVD algorithm with a preconditioning stage and new iterative scheme for preconditioned triangular matrices.

Our results show that the new preconditioned Jacobi-type SVD algorithm can be competitive in efficiency with fast bidiagonalization-based methods (QR and divide-and-conquer algorithms) without trading accuracy for speed. *In fact, the new algorithm computes the SVD more accurately and almost twice as fast as the SGESVD from LAPACK, and the speedup factor over the equally accurate standard one-sided Jacobi SVD (enhanced with de Rijk's pivoting) typically ranges between three and ten.* (See Figures 4.1 and 4.2.) The new Jacobi SVD algorithm remains slower (with factors ranging from 1 to 1.75) than the less accurate divide-and-conquer SVD (SGESDD from LAPACK). If only the singular values are computed, our method can be up to fifteen times faster than the standard one-sided Jacobi SVD, and it is up to twice as slow as bidiagonalization-based singular value computation. (In our tests with dimensions up to 1000 the efficiency was within factors 0.63–2.43 of SGESVD and SGESDD.)

The new algorithm has an advantage over bidiagonalization-based methods in cases of matrices with low numerical rank, because it starts with a rank revealing factorization (cf. [23]). (Bidiagonalization is not rank revealing unless enhanced with pivoting, which would make it more expensive.) In fact, in cases of low numerical rank, we can even outperform SGESDD. If $m \gg n$, both methods start with the QR factorization, which is the most expensive part of the computation, and in that case the efficiency of all three methods is about the same. Also, if only the standard absolute accuracy is required, then Jacobi iterations can be controlled by a loosened stopping criterion, thus allowing satisfactory approximation with less computational effort.

However, our algorithm is not yet fully optimized and several issues are the subject of our current and future work. The most expensive part of our method are BLAS 1 Jacobi rotations, which means that there is more potential for optimization. One simple improvement will be possible once optimized combinations of AXPY and DOT, and combination of two linked AXPY operations are available in single calls.

Nontrivial improvement will be obtained by using block rotations. We expect that the fast scaled block rotations designed by Hari [21] will fully exploit the potential of our approach and considerably improve the performance, especially in difficult cases of tightly clustered singular values. Namely, in such cases the preconditioner does not perform well, which means that one or two full sweeps of rotations are needed to compensate the lack of proper preconditioning. Such heavy computation on a BLAS 1 level is clearly seen in the numerical results presented in section 4 (see, e.g., the crosses above 1.5 and 2 on Figure 4.4).

Further, the overhead of column pivoting in BLAS 3 optimized QR factorization (SGEQP3 from LAPACK) is still too big. Any future improvement of the column-pivoted QR factorization contributes to the efficiency of our algorithm. Other issues include new rank revealing QR factorization and using shifts in the second QR (and third in some cases) factorization if only classical absolute error bounds are required. Our ultimate goal is high performance LAPACK-style software. The question is whether or not we will be able to reach the efficiency of SGESDD. Time will tell.

**Acknowledgments.** The authors acknowledge generous support by the Volkswagen Science Foundation and the Croatian Ministry of Science and Technology. They are also indebted to P. Arbenz (Zürich), J. Barlow (State College), J. Demmel (Berkeley), F. Dopico (Madrid), V. Hari (Zagreb), W. Kahan (Berkeley), J. Moro (Madrid), B. Parlett (Berkeley), and I. Slapničar (Split) for their comments, criticisms, and many fruitful discussions. Special thanks go to the anonymous referees for their substantial and constructive suggestions.

## REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenny, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, 2nd ed., SIAM, Philadelphia, 1995.

[2] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.

[3] C. H. Bischof and G. Quintana-Orti, *Algorithm* 782: *Codes for rank-revealing QR factorizations of dense matrices*, ACM Trans. Math. Software, 24 (1998), pp. 254–257.

[4] C. H. Bischof and G. Quintana-Orti, *Computing rank-revealing QR factorizations of dense*

*matrices*, ACM Trans. Math. Software, 24 (1998), pp. 226–253.

[5] N. N. Chan and K.-H. Li, *Diagonal elements and eigenvalues of a real symmetric matrix*, J. Math. Anal. Appl., 91 (1983), pp. 562–566.

[6] S. Chandrasekaran and I. C. F. Ipsen, *Analysis of a QR algorithm for computing singular values*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 520–535.

[7] P. I. Davies and N. J. Higham, *Numerically stable generation of correlation matrices and their factors*, BIT, 40 (2000), pp. 640–651.

[8] J. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[9] P. P. M. de Rijk, *A one-sided Jacobi algorithm for computing the singular value decomposition on a vector computer*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 359–371.

[10] Z. Drmač, *Computing the Singular and the Generalized Singular Values*, Ph.D. thesis, Lehrgebiet Mathematische Physik, Fernuniversität Hagen, Germany, 1994.

[11] Z. Drmač, *Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic*, SIAM J. Sci. Comput., 18 (1997), pp. 1200–1222.

[12] Z. Drmač, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.

[13] Z. Drmač and Z. Bujanović, *On the failure of rank revealing QR factorization software—a case study*, ACM Trans. Math. Software, to appear.

[14] Z. Drmač and K. Veselić, *New fast and accurate Jacobi SVD algorithm.* I, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1322–1342.

[15] K. V. Fernando and B. N. Parlett, *Implicit Cholesky algorithms for singular values and vectors of triangular matrices*, Numer. Linear Algebra Appl., 2 (1995), pp. 507–531.

[16] A. George, K. Ikramov, and A. B. Kucherov, *Some properties of symmetric quasi-definite matrices*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1318–1323.

[17] H. H. Goldstine, H. H. Murray, and J. von Neumann, *The Jacobi method for real symmetric matrices*, J. Assoc. Comput. Mach., 6 (1959), pp. 59–96. (Also in Collected Works, Vol. V, J. von Neumann, ed., Pergamon Press, New York, 1973, pp. 573–610.)

[18] K. Goto, http://www.cs.utexas.edu/users/kgoto/, 2004.

[19] K. Goto and R. van de Geijn, *On Reducing TLB Misses in Matrix Multiplication*, Tech. report TR-2002-55, FLAME Working Note 9, Department of Computer Science, The University of Texas at Austin, Austin, TX, 2002.

[20] V. Hari, *On sharp quadratic convergence bounds for the serial Jacobi methods*, Numer. Math., 60 (1991), pp. 375–406.

[21] V. Hari, *Accelerating the SVD block-Jacobi method*, Computing, 75 (2005), pp. 27–53.

[22] V. Hari, *Convergence of a block-oriented quasi-cyclic Jacobi method*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 349–369.

[23] V. Hari and K. Veselić, *On Jacobi methods for singular value decompositions*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 741–754.

[24] M. R. Hestenes, *Inversion of matrices by biorthogonalization and related results*, J. Soc. Indust. Appl. Math., 6 (1958), pp. 51–90.

[25] C. G. J. Jacobi, *Über ein leichtes Verfahren die in der Theorie der Säcularstörungen vorkommenden Gleichungen numerisch aufzulösen*, Crelle's Journal für reine und angew. Math., 30 (1846), pp. 51–95.

[26] W. F. Mascarenhas, *On the convergence of the Jacobi method for arbitrary orderings*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1197–1209.

[27] R. Mathias and G. W. Stewart, *A block QR algorithm for singular value decomposition*, Linear Algebra Appl., 182 (1993), pp. 91–100.

[28] N. H. Rhee and V. Hari, *On the global and cubic convergence of a quasi-cyclic Jacobi method*, Numer. Math., 66 (1993), pp. 97–122.

[29] G. W. Stewart, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal., 17 (1980), pp. 403–409.

[30] G. W. Stewart, *Perturbation Theory for the Singular Value Decomposition*, Tech. report UMIACS-TR-90-124, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1990.

[31] G. W. Stewart, *A Gap-Revealing Matrix Decomposition*, Tech. report TR-3771, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1997.

[32] G. W. Stewart, *The QLP Approximation to the Singular Value Decomposition*, Tech. report TR-97-75, Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 1997.

[33] K. Veselić and V. Hari, *A note on a one-sided Jacobi algorithm*, Numer. Math., 56 (1989), pp. 627–633.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1363–1381

# ALGORITHMIC ASPECTS OF ELIMINATION TREES FOR SPARSE UNSYMMETRIC MATRICES*

STANLEY C. EISENSTAT† AND JOSEPH W. H. LIU‡

**Abstract.** The elimination tree of a symmetric matrix plays an important role in sparse elimination. We recently defined a generalization of this structure to the unsymmetric case that retains many of its properties. Here we present an algorithm for constructing the elimination tree of an unsymmetric matrix and show how it can be used to find a symmetric reordering of the matrix into a recursive, bordered block triangular form. We also present two symbolic factorization algorithms that use the elimination tree to determine the nonzero structures of the triangular factors of such matrices. Numerical experiments demonstrate that these algorithms are efficient and compare the new symbolic factorization schemes with existing ones.

**Key words.** elimination tree, symbolic factorization, sparse matrix factorization

**AMS subject classifications.** 65F05, 65F50

**DOI.** 10.1137/050643581

## 1. Introduction.

The elimination tree of a symmetric matrix plays many important roles in sparse factorization [20]. By using paths instead of edges to define the tree, we recently generalized this structure to unsymmetric matrices while retaining many of its properties [15]. In this paper we consider several algorithmic aspects of elimination trees. The outline of the paper is as follows.

In section 2 we introduce some graph notation and review the elimination tree of an unsymmetric matrix and tree-based, bordered block triangular (BBT) orderings [15]. In section 3 we present algorithms for constructing the elimination tree and for finding a BBT ordering. The basic algorithm repeatedly finds strongly connected components of subgraphs. Quotient graphs and path-preserving edge reduction are used to improve its efficiency.

In section 4 we review existing algorithms for symbolic $LU$ factorization, which use either symmetric pruning [13] or the elimination dags (directed acyclic graphs) [18] to improve performance. We describe a tree-based pruning scheme that is the first practical implementation of path-symmetric reduction [13]. We also present two symbolic factorization algorithms for matrices that are in BBT form. Both are based on special structural properties of the triangular factors of BBT ordered matrices and this tree-based pruning scheme.

In section 5 we give experimental results from running these algorithms on a set of test problems. We also compare the performance of the new symbolic factorization codes with existing ones. In section 6 we offer some concluding remarks and suggest possible applications of elimination trees to other aspects of sparse factorization.

---

*Received by the editors October 26, 2005; accepted for publication (in revised form) by E. Ng July 2, 2007; published electronically January 4, 2008.

http://www.siam.org/journals/simax/29-4/64358.html

†Department of Computer Science, Yale University, P.O. Box 208285, New Haven, CT 06520-8285 (stanley.eisenstat@yale.edu).

‡Department of Computer Science, York University, North York, ON, M3J 1P3, Canada (joseph@cs.yorku.ca). This author's research was supported in part by the Natural Sciences and Engineering Research Council of Canada under grant A5509.

$$A = \begin{pmatrix} 1 & & & \bullet & & & & & & \\ \bullet & 2 & \bullet & & & & & & \bullet & \bullet \\ & & 3 & & \bullet & \bullet & & & \bullet & \\ & & & 4 & & & \bullet & & & \\ & \bullet & & & 5 & & & & & \\ \bullet & & & & & 6 & & & & \bullet \\ & & & & & \bullet & 7 & & & \\ & & & & & \bullet & & 8 & & \\ \bullet & \bullet & & & & & & & 9 & \\ & & \bullet & & & & & \bullet & & 10 \end{pmatrix}$$

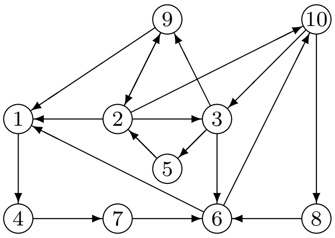


FIG. 1. *A sparse matrix and its directed graph.*

## 2. Background.

The authors introduced the elimination tree of a sparse unsymmetric matrix and studied tree-based, BBT orderings in [15]. We briefly review these ideas here.

### 2.1. Graph notation.

The *directed graph* $G(M)$ of a sparse unsymmetric $n \times n$ matrix $M$ is defined as follows: the vertex set is $X(M) = \{1, 2, \ldots, n\}$; and for distinct vertices $r$ and $c$, there is a *directed edge* $r \xmapsto{M} c$ from $r$ to $c$ if and only if the entry $m_{rc} \neq 0$. We shall use the notation $r \overset{M}{\Longrightarrow} c$ to indicate a *directed path*[1] from $r$ to $c$. If the matrix $M$ is clear from context, we shall sometimes use the abbreviated forms $r \mapsto c$ and $r \Rightarrow c$.

At times we shall consider composite paths such as $r \xmapsto{M} u \overset{M'}{\Longrightarrow} c$. If there is no restriction on the intermediate vertex $u$, we shall use the abbreviated form[2] $r \xmapsto{M} \overset{M'}{\Longrightarrow} c$.

A set of vertices $S \subseteq X(M)$ induces a *subgraph* of $G(M)$ consisting of the vertices in $S$ and all edges $u \mapsto v$ with $u, v \in S$. To simplify the presentation we shall not distinguish between a set of vertices and the subgraph of $G(M)$ that it induces; that is, we shall use $S$ as a subset of $X(M)$ and as a subgraph of $G(M)$ interchangeably. It should be clear from context which use is intended. We shall use the notation $G_m(M)$ to denote the subgraph $\{1, 2, \ldots, m\}$ of $G(M)$ for $0 \leq m \leq n$.

A subgraph $S$ of $G(M)$ is *strongly connected* if, for any pair of distinct vertices $u, v \in S$, there is a cycle $u \Rightarrow v \Rightarrow u$ in the subgraph $S$. If $S$ is a *maximal* strongly connected subgraph, then it is a (strongly connected) *component* of $G(M)$.

Figure 1 contains a $10 \times 10$ unsymmetric matrix that will be used as an example throughout the paper. The diagonal entries are the vertices; each $\bullet$ represents an off-diagonal nonzero. The subgraph $\{1, 4, 6, 7\}$ is strongly connected (since $1 \mapsto 4 \mapsto 7 \mapsto 6 \mapsto 1$ is a cycle), but is not a component (since the entire graph is strongly connected).

### 2.2. Quotient graphs and matrices.

Let $\{S_1, S_2, \ldots, S_q\}$ be a partition of the vertex set $X(M)$. The *quotient graph* of $G(M)$ induced by this partition has the vertices $S_1$, $S_2$, $\ldots$, $S_q$ and a directed edge from $S_i$ to $S_j$ if and only if there are vertices $u \in S_i$ and $v \in S_j$ with $u \xmapsto{M} v$. We can represent the quotient graph as $G(Q)$ by defining a $q \times q$ *quotient matrix* $Q$ with $q_{ij} = 1$ if there is an edge from $S_i$ to $S_j$ and $q_{ij} = 0$ otherwise.

---

[1] Paths and cycles must contain at least one edge, but need not be *simple*; that is, they may visit a vertex or edge more than once.

[2] This notation is due to John Gilbert.

$$Q = \begin{pmatrix} S_1 & \bullet & & \bullet & \bullet \\ & S_2 & & & \bullet \\ & \bullet & S_3 & & \\ \bullet & \bullet & & S_4 & \\ \bullet & & \bullet & & S_5 \end{pmatrix}$$

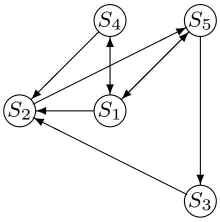


FIG. 2. *A quotient graph of the matrix in Figure* 1.

$$A^+ = \begin{pmatrix} 1 & & & \bullet & & & & & & \\ \bullet & 2 & \bullet & \circ & & & & & \bullet & \bullet \\ & & 3 & & \bullet & \bullet & & & \bullet & \\ & & & 4 & & & \bullet & & & \\ & \bullet & \circ & \circ & 5 & \circ & \circ & & \circ & \circ \\ \bullet & & & \circ & & 6 & \circ & & & \bullet \\ & & & & & \bullet & 7 & & & \circ \\ & & & & & \bullet & \circ & 8 & & \circ \\ \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & & 9 & \circ \\ & & \bullet & & \circ & \circ & \circ & \bullet & \circ & 10 \end{pmatrix}$$

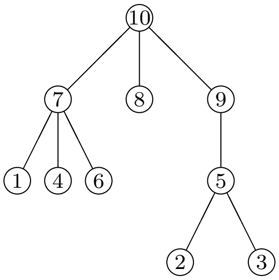


FIG. 3. *The filled matrix and elimination tree for the matrix of Figure* 1.

Figure 2 contains the quotient graph $G(Q)$ of the directed graph in Figure 1 for the partition

$$S_1 = \{2,3,5\}, \quad S_2 = \{1,4,6,7\}, \quad S_3 = \{8\}, \quad S_4 = \{9\}, \quad S_5 = \{10\}.$$

Note that there is an edge from $S_3$ to $S_2$ since there is an edge $8 \mapsto 6$ in the original graph. Also, there is a directed edge from $S_1$ to $S_2$ since there is an edge $2 \mapsto 1$ (or an edge $3 \mapsto 6$).

### 2.3. The elimination tree of an unsymmetric matrix.

Let $A$ be a nonsingular sparse unsymmetric $n \times n$ matrix with a nonzero diagonal and the factorization $A = LU$, where $L$ is unit lower triangular and $U$ is upper triangular; and let $A^+ = L + U - I$ denote its *filled matrix.*

We define the *elimination tree* $T(A)$ of $A$ using the parent function

$$\textsc{fpnz}(k) = \min\{x \mid x > k \text{ and } x \overset{L}{\Longrightarrow} k \overset{U}{\Longrightarrow} x\},$$

where the minimum over the empty set is taken to be $\infty$. Vertex $p = \textsc{fpnz}(k)$ is the parent of vertex $k$ if $\textsc{fpnz}(k) < \infty$. By definition we have $\textsc{fpnz}(n) = \infty$, so that vertex $n$ is a root. In general $T(A)$ consists of one or more trees, and each vertex $k$ with $\textsc{fpnz}(k) = \infty$ is a root. However, there is only one tree when $A$ is irreducible [15, Corollary 3.7].

For example, the filled matrix and elimination tree for the matrix in Figure 1 is given in Figure 3. The diagonal entries are the vertices; each $\circ$ represents an off-diagonal nonzero in $A^+$ that is zero in $A$.

We shall use the notation $\mathcal{T}_k$ to denote both the subtree of $T(A)$ rooted at vertex $k$ and the set of vertices in this subtree.

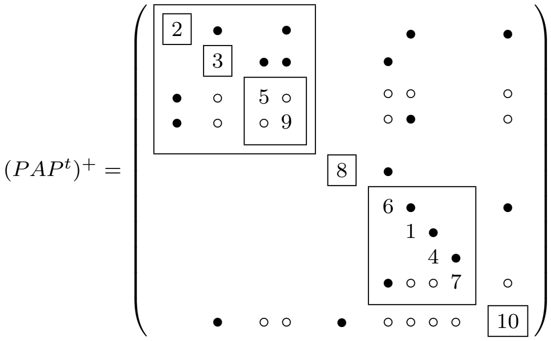


FIG. 4. *The filled matrix of an upper BBT postordering of the matrix in Figure* 1.

**2.4. BBT postorderings.** A matrix $M$ is said to be upper *bordered block triangular* (BBT) if it can be written in the form

$$\begin{pmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,s} & M_{1,s+1} \\ 0 & M_{2,2} & \cdots & M_{2,s} & M_{2,s+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & M_{s,s} & M_{s,s+1} \\ M_{s+1,1} & M_{s+1,2} & \cdots & M_{s+1,s} & M_{s+1,s+1} \end{pmatrix}$$

for some symmetric blocking of the rows and columns. If we can find a permutation matrix $P$ such that the reordered matrix $PAP^t$ is in BBT form for some $s > 1$, we can take advantage of this structure by working with $PAP^t$ instead of $A$.

Let $c_1, \ldots, c_t$ be the children of a vertex $k$ in the elimination tree $T(A)$. In a *postordering* of $T(A)$ the vertices within each subtree $\mathcal{T}_{c_i}$ are numbered consecutively, and $k$ is numbered immediately after its descendants. Postordering preserves[3] the structure of $T(A)$.

To achieve a (recursive) BBT form, an *upper BBT* (resp., *lower BBT*) postordering imposes a further condition on the order in which the subtrees $\mathcal{T}_{c_1}, \ldots, \mathcal{T}_{c_t}$ are numbered: an edge in $G(A)$ from a vertex in one subtree to a vertex in another must always be directed from the lower-numbered vertex to the higher-numbered one (resp., higher to lower). A BBT postordering need not be unique and need not preserve the filled graph.[4] However, it does preserve the set of values on the diagonal of $U$ (see [15, Theorem 5.2]) and thus is arguably just as stable numerically.

For example, the ordering $2, 3, 5, 9, 8, 6, 1, 4, 7, 10$ is an upper BBT postordering of the elimination tree in Figure 3 and gives rise to the permuted matrix $PAP^t$ in Figure 4. The boxes emphasize the recursive nature of BBT form.

We shall say that a matrix $A$ is upper (resp., lower) *BBT ordered* if the natural ordering is an upper (resp., lower) BBT postordering of $T(A)$.

## 3. Constructing the elimination tree.

**3.1. The basic algorithm.** In this section we consider effective ways to construct the elimination tree $T(A)$. All are based on the following result, which is a

---

[3] That is, letting $P$ denote the corresponding permutation matrix, the elimination tree $T(PAP^t)$ is identical to $T(A)$ up to the numbering of the vertices [15, Theorem 5.1].

[4] Indeed, it may increase the amount of fill.

TABLE 1
*Strongly connected components of $G_k(A)$.*

| $k$ | Components | $k$ | Components |
|---|---|---|---|
| 1 | {1} | 6 | {1}, {4}, {2,3,5}, {6} |
| 2 | {1}, {2} | 7 | {2,3,5}, {1,4,6,7} |
| 3 | {1}, {2}, {3} | 8 | {2,3,5}, {1,4,6,7}, {8} |
| 4 | {1}, {2}, {3}, {4} | 9 | {1,4,6,7}, {8}, {2,3,5,9} |
| 5 | {1}, {4}, {2,3,5} | 10 | {1,2,3,4,5,6,7,8,9,10} |

```
Algorithm eTree
  for vertex $k = 1$ to $n$ do
    Find the component $\mathcal{K}$ of $G_k(A)$ that contains $k$
    for each vertex $x \in \mathcal{K} \setminus \{k\}$ do
      if fpnz($x$) $= \infty$ then fpnz($x$) $= k$
    end for
    fpnz($k$) $= \infty$
  end for
```

FIG. 5. *Basic algorithm for constructing the elimination tree.*

more usable characterization of the parent function FPNZ($*$) than the definition given in section 2.3.

THEOREM 3.1 (see [15, Corollary 3.4]). *Vertex $k$ is the parent of vertex $x$ in the elimination tree $T(A)$ if and only if $k$ is the first vertex after $x$ such that $k$ and $x$ belong to the same strongly connected component of the subgraph $G_k(A)$ of $G(A)$.*

For example, Table 1 gives the strongly connected components in $G_k(A)$ for the matrix in Figure 1, with the component containing $k$ listed last. From this it is easy to verify that the elimination tree is as shown in Figure 3. For example, vertex 5 is the parent of vertex 2 because $G_5(A)$ is the first subgraph $G_k(A)$ with a component containing vertices 2 and $k$.

Theorem 3.1 leads immediately to Algorithm ETREE in Figure 5, which sets FPNZ($x$) equal to the first $k$ for which $x$ and $k$ belong to the same component of $G_k(A)$.

There are many algorithms for finding strongly connected components (e.g., [7, p. 489], [17], and [23]). When adapted to find the component containing $k$, all take time proportional to the number of vertices and edges in $G_k(A)$ reachable from $k$. Thus Algorithm ETREE runs in time $O(mn)$, where $m$ is the number of nonzero entries in $A$.

There is still benefit to reducing both the number of these vertices (see section 3.2) and the number of these edges (see section 3.3) in such a way that strong connectivity is preserved.

**3.2. Using quotients of strongly connected components.** The strongly connected components of a graph form a partition of its vertex set, and by definition the quotient graph they induce has the following property:

> Let distinct vertices $x$ and $y$ belong to strongly connected components $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then there exists a path $x \Rightarrow y$ in the graph if and only if either $\mathcal{X} = \mathcal{Y}$ or there exists a path $\mathcal{X} \Rightarrow \mathcal{Y}$ in the quotient graph.

Moreover, the quotient graph is *acyclic* (i.e., cycle-free). In this section we use this succinct representation of the connectivity of a graph [21] to improve Algorithm eTree. We begin with a characterization of the components of the graph $G_k(A)$.

Theorem 3.2. *Let $\mathcal{K}$ be a component of $G_k(A)$, and let $m$ be the highest-numbered vertex in $\mathcal{K}$. Then $\mathcal{K} = \mathcal{T}_m$, the subtree of the elimination tree $T(A)$ rooted at $m$.*

*Proof.* ($\mathcal{K} \subseteq \mathcal{T}_m$) Assume that $\mathcal{K} \not\subseteq \mathcal{T}_m$, and let $x$ be the highest-numbered vertex in $\mathcal{K} \setminus \mathcal{T}_m$. Since $\mathcal{K}$ is a component of $G_k(A)$ and $m$ is the highest-numbered vertex in $\mathcal{K}$, it follows that $\mathcal{K}$ is a component of $G_m(A)$. Since in addition both $x$ and $m$ belong to $\mathcal{K}$, by Theorem 3.1 vertex $x$ has a parent $p$ in $T(A)$, both $x$ and $p$ belong to the same component of $G_p(A)$, and $x < p \leq m$. Thus both $x$ and $p$ belong to the same component of $G_m(A)$, namely, $\mathcal{K}$. On the other hand, since $x \notin \mathcal{T}_m$, we have $p \notin \mathcal{T}_m$ so that $p \in \mathcal{K} \setminus \mathcal{T}_m$, which contradicts the definition of $x$.

($\mathcal{T}_m \subseteq \mathcal{K}$) Since $\mathcal{T}_m$ is the subtree of $T(A)$ rooted at $m$ and $m \in \mathcal{K}$, it suffices to show that if vertex $p \in \mathcal{K}$ and vertex $c$ is a child of $p$ in $T(A)$, then $c \in \mathcal{K}$. By Theorem 3.1 both $p$ and $c$ belong to the same component of $G_p(A)$, which is a subgraph of $G_k(A)$ since $p \leq k$. Thus $c$ and $p$ must also belong to the same component of $G_k(A)$, namely, $\mathcal{K}$. ☐

For example, the subgraph $G_7(A)$ of the graph in Figure 1 has two strongly connected components $\{2, 3, 5\}$ and $\{1, 4, 6, 7\}$ (see Table 1) corresponding to the subtrees $\mathcal{T}_5$ and $\mathcal{T}_7$ of the elimination tree in Figure 3.

Let $G(Q_k)$ denote the quotient graph of $G_k(A)$ induced by its strongly connected components, where $Q_k$ is the quotient matrix. Then each vertex in $G(Q_k)$ is a subtree $\mathcal{T}_u$ of $T(A)$ for some vertex $u$ with $u \leq k$; and for distinct vertices $\mathcal{T}_u$ and $\mathcal{T}_v$ in $G(Q_k)$, there is an edge $\mathcal{T}_u \mapsto \mathcal{T}_v$ if and only if there exist vertices $x \in \mathcal{T}_u$ and $y \in \mathcal{T}_v$ such that $x \overset{A}{\longmapsto} y$.

The graph $G_k(A)$ is obtained from the graph $G_{k-1}(A)$ (the empty graph if $k = 1$) by adding vertex $k$ and all edges $x \overset{A}{\longmapsto} k$ and $k \overset{A}{\longmapsto} x$ with $1 \leq x < k$. Similarly, $G(Q_k)$ can be obtained from $G(Q_{k-1})$ (the empty graph if $k = 1$) in two steps:

1. Form the intermediate graph $G(Q'_k)$ from $G(Q_{k-1})$ by
   - adding vertex $k$;
   - for each vertex $\mathcal{T}_u$, adding the edge $\mathcal{T}_u \mapsto k$ if there is an edge $x \overset{A}{\longmapsto} k$ with $x \in \mathcal{T}_u$; and
   - for each vertex $\mathcal{T}_v$, adding the edge $k \mapsto \mathcal{T}_v$ if there is an edge $k \overset{A}{\longmapsto} y$ with $y \in \mathcal{T}_v$.
2. Form $G(Q_k)$ from $G(Q'_k)$ by coalescing $k$ and all vertices in the strongly connected component of $G(Q'_k)$ that contains it into the new vertex $\mathcal{T}_k$.

Figure 6 gives the intermediate quotient graphs $G(Q'_7)$, $G(Q'_8)$, $G(Q'_9)$, and $G(Q'_{10})$ for the matrix in Figure 1. For example, the four components of $G_6(A)$,

$$\mathcal{T}_1 = \{1\}, \qquad \mathcal{T}_4 = \{4\}, \qquad \mathcal{T}_5 = \{2, 3, 5\}, \qquad \mathcal{T}_6 = \{6\},$$

are used to form $G(Q'_7)$.

Let $\mathcal{T}_c$ be a vertex of $G(Q'_k)$ that is coalesced into $\mathcal{T}_k$, and let vertex $p$ be the parent of vertex $c$ in $T(A)$. Then $c$ and $k$ belong to the same component of $G_k(A)$, and $p \leq k$ by Theorem 3.1. But if $p < k$, then $\mathcal{T}_c$ could not have been a component of $G_{k-1}(A)$ or a vertex of $G(Q_{k-1})$ by Theorem 3.2. Thus $p = k$, and $c$ is a child of $k$. This leads immediately to Algorithm eTreeQ in Figure 7.

$$Q'_7 = \left(\begin{array}{cccc|c} \mathcal{T}_1 & \bullet & & & \\ & \mathcal{T}_4 & & & \bullet \\ \bullet & & \mathcal{T}_5 & \bullet & \\ \bullet & & & \mathcal{T}_6 & \\ \hline & & & \bullet & 7 \end{array}\right)$$

$$Q'_8 = \left(\begin{array}{cc|c} \mathcal{T}_5 & \bullet & \\ & \mathcal{T}_7 & \\ \hline & \bullet & 8 \end{array}\right)$$

$$Q'_9 = \left(\begin{array}{ccc|c} \mathcal{T}_5 & \bullet & & \bullet \\ & \mathcal{T}_7 & & \\ & \bullet & \mathcal{T}_8 & \\ \hline \bullet & \bullet & & 9 \end{array}\right)$$

$$Q'_{10} = \left(\begin{array}{ccc|c} \mathcal{T}_7 & & & \bullet \\ \bullet & \mathcal{T}_8 & & \\ \bullet & & \mathcal{T}_9 & \bullet \\ \hline & \bullet & \bullet & 10 \end{array}\right)$$

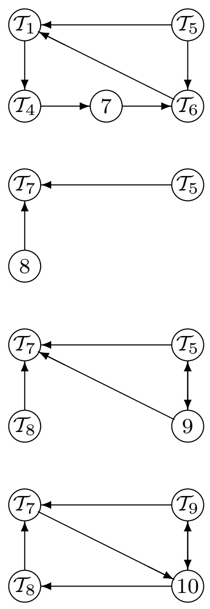


Fig. 6. *Some of the intermediate quotient graphs for the matrix in Figure* 1. *The leading diagonal block of* $Q'_k$ *is* $Q_{k-1}$.

```
Algorithm eTreeQ
    $G(Q_0)$ = empty graph
    for vertex $k = 1$ to $n$ do
        Create $G(Q'_k)$ from $G(Q_{k-1})$ by adding $k$ and
            its incident edges
        Find the component $\mathcal{K}$ of $G(Q'_k)$ that contains $k$
        for each vertex $\mathcal{T}_c \in \mathcal{K} \setminus \{k\}$ do
            FPNZ($c$) = $k$
        end for
        Create $G(Q_k)$ from $G(Q'_k)$ by coalescing
            the vertices in $\mathcal{K}$ into the new vertex $\mathcal{T}_k$
        FPNZ($k$) = $\infty$
    end for
```

Fig. 7. *Constructing the elimination tree using quotient graphs.*

To find the component of $G(Q'_k)$ containing $k$ we could adapt any of the standard algorithms for finding strongly connected components. However, $G(Q'_k)$ has the property that, when $k$ and its incident edges are deleted, the subgraph $G(Q_{k-1})$ remaining is acyclic. Algorithm SCC of Figure 8 is a new, special-purpose algorithm that takes advantage of this property. As a result it is much simpler than the other algorithms, which work for general graphs. The following result proves its correctness.

Theorem 3.3. *Algorithm* SCC *returns the strongly connected component of* $G(Q'_k)$ *that contains vertex* $k$.

*Proof.* Since there is a path $k \Rightarrow v$ in $G(Q'_k)$ for each vertex $v$ visited during the depth-first search DFS($k$), it suffices to prove that INSC$[v]$ = **true** when DFS($v$) returns if and only if there is a path $v \Rightarrow k$. We use induction on the order in which DFS($v$) returns.

From the algorithm we see that for any $v \neq k$, we have INSC$[v]$ = **true** if and only if INSC$[w]$ = **true** for some neighbor $w$ of $v$. Since either $w = k$ or DFS($w$)

```
Algorithm SCC (vertex $k$)
    for each vertex $v$ in $G(Q'_k)$ do
        VISITED[$v$] = INSC[$v$] = false
    end for
    INSC[$k$] = true;   SC = {$k$}
    DFS($k$)
    return SC

    procedure DFS (vertex $v$)
        VISITED[$v$] = true
        for each vertex $w$ adjacent to $v$ in $G(Q'_k)$ do
            if not VISITED[$w$] then DFS($w$)
            if INSC[$w$] then INSC[$v$] = true
        end for
        if INSC[$v$] then SC = SC ∪ {$v$}
    end procedure
```

FIG. 8. *Finding the strongly connected component of $G(Q'_k)$ that contains vertex $k$.*

returns before DFS($v$), by induction the latter is true if and only if either $w = k$ or there is a path $w \Rightarrow k$; i.e., there is a path $v \Rightarrow k$. ☐

The time to initialize the arrays VISITED[∗] and INSC[∗] at the start of Algorithm SCC need not be proportional to the number of vertices and edges reached during the depth-first search. However, suppose that we set all elements in these arrays to 0 at the start of Algorithm ETREEQ and adopt the convention that **true** means "equal to $k$" during the call to SCC($k$). Since none of these elements can have that value before the call, they all have a value equivalent to **false**, and no further initialization is necessary.

While in practice Algorithm ETREEQ is much faster than Algorithm ETREE, its worst-case run-time is still $O(mn)$.

### 3.3. Using path-preserving edge reductions.

Let $u$ and $v$ be distinct vertices in $G(Q_k)$ for which there is both an edge $u \mapsto v$ and a path $u \mapsto\Rightarrow v$ of length at least two. Then the edge $u \mapsto v$ can be deleted without changing the connectivity.[5] Pruning such edges while transforming $G(Q_{k-1})$ into $G(Q_k)$ will improve the performance of Algorithm ETREEQ during subsequent iterations since there will be fewer edges to examine. However, it is not necessary to delete all redundant edges; indeed, it can be more efficient to prune only an easily identifiable subset of them [13].

In this section we shall describe two forms of such *path-preserving edge pruning* that are easily incorporated into Algorithm SCC. We need the following result to justify this approach.

THEOREM 3.4. *Let $W$ be a set of edges in the graph $G(Q'_k)$ with the property that, for each edge $u \mapsto v$ in $W$, there is a corresponding path $u \mapsto\Rightarrow v$ of length at least two that does not contain an edge of the form $x \mapsto k$. Then we can prune all of the edges in $W$ without changing the connectivity.*

*Proof.* Let $G(Q''_k)$ be the graph obtained from $G(Q'_k)$ by deleting every edge in $W$. It suffices to show that for each edge $u \mapsto v$ deleted, there is a path $u \Rightarrow v$ in $G(Q''_k)$. By assumption there is such a path in $G(Q'_k)$. If no edge on this path is in $W$, we are done. Otherwise replace the first such edge $x \mapsto y$ by the corresponding path $x \Rightarrow y$ in $G(Q'_k)$ and repeat. Since no replacement path contains an edge of the

---

[5] These transformations were used in [14] to define effective implicit representations of the nonzero structure of the Schur complement of a sparse matrix.

form $x \mapsto k$ and every cycle in $G(Q'_k)$ must contain such an edge, none of the paths created can contain a cycle and this process must eventually halt. $\square$

For example, in the graph $G(Q'_7)$ of Figure 6 the set with the single edge $\mathcal{T}_5 \mapsto \mathcal{T}_1$ satisfies the condition in the theorem since there is a path $\mathcal{T}_5 \mapsto \mathcal{T}_6 \mapsto \mathcal{T}_1$ of length two, but the set with the single edge $\mathcal{T}_5 \mapsto \mathcal{T}_6$ does not since the only path $\mathcal{T}_5 \mapsto \Rightarrow \mathcal{T}_6$ of length at least two contains the edge $\mathcal{T}_4 \mapsto 7$.

The first form of pruning applies to edges emanating from $k$. Let $\text{ISPATH}[w] =$ **true** if and only if there is a path $k \Rightarrow v \mapsto w$ in $G(Q'_k)$ for some vertex $v \neq k$. Then the set of edges

$$W_1 = \{k \mapsto w \mid \text{ISPATH}[w]\}$$

is redundant. We can compute the $\text{ISPATH}[*]$ array by initializing it to **false** and inserting the statement

$$\textbf{if } v \neq k \textbf{ then } \text{ISPATH}[w] = \textbf{true}$$

into the **for each** loop in function $\text{DFS}(*)$.

The second form of pruning applies to *forward* edges. Let $\text{PRE}[u] = i$ if $u$ is the $i$th vertex visited during the depth-first search. Assume that the call $\text{DFS}(v)$ finds a vertex $w$ adjacent to $v$ that has already been visited (i.e., $\text{VISITED}[w] =$ **true**). By definition, if $\text{PRE}[w] > \text{PRE}[v]$, then $w$ must have been visited after the call $\text{DFS}(v)$ began. Moreover, that visit must have been the result of a call $\text{DFS}(x)$ for some other vertex $x$ that is adjacent to $v$; i.e., $w$ must be on a path $v \mapsto x \Rightarrow w$ of length of least two. Thus the set of forward edges

$$W_2 = \{v \mapsto w \mid \text{PRE}[w] > \text{PRE}[v] \text{ but } w \text{ was not visited from } v\}$$

is redundant. We can compute the $\text{PRE}[*]$ array by initializing an integer value $\text{DFSNO}$ to 0 and inserting the statements

$$\text{PRE}[v] = \text{DFSNO}; \quad \text{DFSNO} = \text{DFSNO} + 1$$

at the start of function $\text{DFS}(*)$.

A version of Algorithm SCC that incorporates both forms of pruning is shown in Figure 9. It runs somewhat faster but has the same worst-case complexity.

### 3.4. Finding a BBT postordering.

Once we have constructed the elimination tree we can find a postordering using a depth-first search where each vertex $k$ is numbered after all of its children have been visited. However, a BBT postordering imposes an additional constraint on the *order* in which the children are visited. In this section we show that Algorithm SCC provides this order.

Let vertices $c_1, \ldots, c_t$ be the children of $k$ in $T(A)$. By Theorem 3.2 the subtrees $\mathcal{T}_{c_1}, \ldots, \mathcal{T}_{c_t}$ of $T(A)$ are the strongly connected components of $G_{k-1}(A)$ that are coalesced into the component $\mathcal{T}_k$ of $G_k(A)$ during the formation of $G(Q_k)$. The additional condition for a lower BBT postordering is that $c_i$ must be numbered after $c_j$ if there is an edge $x \mapsto y$ with $x \in \mathcal{T}_{c_i}$ and $y \in \mathcal{T}_{c_j}$. In terms of the quotient graph $G(Q'_k)$, this means that $c_i$ must be numbered after $c_j$ if there is an edge $\mathcal{T}_{c_i} \mapsto \mathcal{T}_{c_j}$.

Consider the order in which vertices are added to $\mathcal{K}$ in Algorithm SCC. Vertex $k$ is added first; and if there is an edge $\mathcal{T}_{c_i} \mapsto \mathcal{T}_{c_j}$ in $G(Q'_k)$, then vertex $\mathcal{T}_{c_i}$ is added to $\mathcal{K}$ after vertex $\mathcal{T}_{c_j}$. That is, ignoring vertex $k$, this order satisfies the condition for a lower BBT postordering. Similarly, the reverse of this order satisfies the condition for an upper BBT postordering.

```
Algorithm sccP (vertex $k$)
    for each vertex $v$ in $G(Q'_k)$ do
        visited[$v$] = inSC[$v$] = isPath[$v$] = false
    end for
    inSC[$k$] = true;   SC = $\{k\}$;   dfsNo = 0
    dfsP($k$)
    for each vertex $v$ adjacent to $k$ do
        if isPath[$v$] then prune $k \mapsto v$
    end for
    return SC

    procedure dfsP (vertex $v$)
        visited[$v$] = true
        pre[$v$] = dfsNo;   dfsNo = dfsNo + 1
        for each vertex $w$ adjacent to $v$ in $G(Q'_k)$ do
            if not visited[$w$] then dfsP($w$)
            else if pre[$w$] > pre[$v$] then prune $v \mapsto w$
            if $v \neq k$ then isPath[$w$] = true
            if inSC[$w$] then inSC[$v$] = true
        end for
        if inSC[$v$] then SC = SC $\cup \{v\}$
    end procedure
```

FIG. 9. *Finding the strongly connected component using edge pruning.*

## 4. Symbolic *LU* factorization.

Symbolic factorization is the process of determining the nonzero structure of the triangular factors $L$ and $U$. We can find the structure of $L$ either by rows or by columns and the structure of $U$ either by rows or by columns. In this section we shall discuss the by-row variants. We assume that no pivoting is required for numerical stability.

Let $\mathrm{Struct}(M_{r*})$ denote the nonzero structure of the $r$th row of the matrix $M$, that is, the vertex set $\{j \mid m_{rj} \neq 0\}$. We can characterize $\mathrm{Struct}(L_{r*})$ and $\mathrm{Struct}(U_{r*})$ using the nonzero structure of the original matrix and paths in the graphs $G(L)$ and $G(U)$ of the factor matrices:

$$\mathrm{Struct}(L_{r*}) = \{\, j \mid j \leq r \text{ and either } r \xmapsto{A} j \text{ or } r \xmapsto{A} \xRightarrow{U_{RR}} j\},$$

where $U_{RR}$ denotes the $r \times r$ leading principal submatrix of $U$ (see [22, Theorem 6]), and

$$\mathrm{Struct}(U_{r*}) = \left( \mathrm{Struct}(A_{r*}) \cup \bigcup_{k \in \mathrm{Struct}(L_{r*}) \setminus \{r\}} \mathrm{Struct}(U_{k*}) \right) \setminus \{1, \dots, r-1\}$$

(see [6]).

### 4.1. Pruning.

Reducing the number of edges in $G(U_{RR})$ that must be traversed to find $\mathrm{Struct}(L_{r*})$ and the number of vertices $k \in \mathrm{Struct}(L_{r*}) \setminus \{r\}$ for which $\mathrm{Struct}(U_{k*})$ contributes to $\mathrm{Struct}(U_{r*})$ will speed up symbolic factorization. We can do both by using path-preserving edge pruning as discussed in section 3.3.

Let $G(\bar{U})$ be the subgraph of $G(U)$ obtained by pruning some set of redundant edges, that is, edges $x \xmapsto{U} y$ for which there is a path $x \xmapsto{U} \xRightarrow{U} y$. Here $\bar{U}$ is the matrix obtained from $U$ by setting the entries associated with these edges to zero. The graph $G(\bar{L})$ and the matrix $\bar{L}$ are defined similarly.

THEOREM 4.1 (see [13, Observations 3.1 and 3.2]). *The row structure of* $L_{r*}$ *is given by*

$$\mathrm{Struct}(L_{r*}) = \{\, j \mid j \le r \textit{ and either } r \xmapsto{A} j \textit{ or } r \xmapsto{A} \overset{U^{-}_{RR}}{\Longrightarrow} j\};$$

*and the row structure of* $U_{r*}$ *is given by*

$$\mathrm{Struct}(U_{r*}) = \left(\mathrm{Struct}(A_{r*}) \;\cup \bigcup_{k \in \mathrm{Struct}(L^{-}_{r*})\setminus\{r\}} \mathrm{Struct}(U_{k*})\right) \setminus \{1, \ldots, r-1\}.$$

The minimal pruned graphs that can be used in Theorem 4.1 are the *transitive reductions*[6] of $G(L)$ and $G(U)$, also known as the *elimination dags* $G(L^o)$ and $G(U^o)$, respectively [18]. However, the expense[7] of computing these dags could outweigh the savings [13].

A practical compromise is to prune only an easily identifiable subset of edges. Let the function $\rho(k)$ have the property that all edges $i \xmapsto{L} k$ and $k \xmapsto{U} i$ with $i > \rho(k)$ are redundant. Given $\rho(k)$, these edges are easily pruned from $G(L)$ and $G(U)$. Several such *symmetric pruning* functions have been proposed:

- *Symmetric reduction* [13, Theorem 3.3]:

  $$\rho_s(k) = \min\{i \mid i > k \text{ and } i \xmapsto{L} k \xmapsto{U} i\}.$$

  This usually gave the best performance for the matrices considered in [13].
- *Path-symmetric reduction* [13, Theorem 5.1]:

  $$\rho_p(k) = \min\{i \mid i \overset{L}{\Longrightarrow} k \overset{U}{\Longrightarrow} i\}.$$

  This is a generalization of symmetric reduction where the symmetric edges can be paths. No implementation is given in [13]. However, since $\rho_p(k) \equiv$ FPNZ($k$), we can easily evaluate $\rho_p(k)$ by constructing the elimination tree. We shall use $L^\theta$ and $U^\theta$ to represent the reduced factors.
- *Partial path-symmetric reduction* [13, Corollaries 5.2 and 5.3]:

  $$\rho_\pi(k) = \min\{i \mid i \xmapsto{L} k \overset{U}{\Longrightarrow} i \text{ or } i \overset{L}{\Longrightarrow} k \xmapsto{U} i\}.$$

  This is a special case of path-symmetric reduction where one of the symmetric paths must be an edge. It was introduced in [13] as an easily implemented approximation.

Figure 10 gives the factors $L$ and $U$ for the matrix in Figure 1 with the entries pruned by symmetric reduction, partial path-symmetric reduction, path-symmetric reduction, and the elimination dag marked.

**4.2. BBT ordered matrices.** In this section we consider the case where $A$ is upper BBT ordered. The elimination dag $G(L^o)$ is the same as the elimination tree $T(A)$ with edges directed from parent to child [15, Theorem 6.3]. Together with Theorem 4.1, this gives the following result.

---

[6] The *transitive reduction* of a directed graph $G(M)$ is a subgraph $G(M^o)$ such that $u \overset{M}{\Longrightarrow} v$ if and only if $u \overset{M^o}{\Longrightarrow} v$, and that no subgraph with fewer edges has this property. The transitive reduction of a *directed acyclic graph* (dag) is unique [1].

[7] Using supernodes reduces the cost somewhat [19], but the same savings accrue from using pruning as described in what follows.

$$A^+ = \left(\begin{array}{cccccccccc}
1 & & & \bullet & & & & & & \\
\bullet & 2 & \bullet & \circ & & & & & \bullet & s \\
 & & 3 & & \bullet & s & & & s & \\
 & & & 4 & & & \bullet & & & \\
 & \bullet & \circ & \circ & 5 & \circ & \circ & & \circ & s \\
\bullet & & & \circ & & 6 & \circ & & & s \\
 & & & & & \bullet & 7 & & & \circ \\
 & & & & & s & \circ & 8 & & \circ \\
\bullet & \bullet & s & \circ & \circ & s & \circ & & 9 & \circ \\
 & & s & & s & s & \circ & \bullet & \circ & 10
\end{array}\right) = \left(\begin{array}{cccccccccc}
1 & & & \bullet & & & & & & \\
\bullet & 2 & \bullet & \circ & & & & & \bar{\pi} & \pi \\
 & & 3 & & \bullet & \pi & & & \pi & \\
 & & & 4 & & & \bullet & & & \\
 & \bullet & \circ & \circ & 5 & \circ & \circ & & \circ & \pi \\
\bullet & & & \circ & & 6 & \circ & & & \pi \\
 & & & & & \bullet & 7 & & & \circ \\
 & & & & & \pi & \circ & 8 & & \circ \\
\bullet & \bar{\pi} & \pi & \bar{\pi} & \circ & \pi & \circ & & 9 & \circ \\
 & & \pi & & \pi & \pi & \circ & \bullet & \circ & 10
\end{array}\right)$$

$$= \left(\begin{array}{cccccccccc}
1 & & & \bullet & & & & & & \\
\bullet & 2 & \bullet & \circ & & & & & p & p \\
 & & 3 & & \bullet & p & & & p & \\
 & & & 4 & & & \bullet & & & \\
 & \bullet & \circ & \circ & 5 & \circ & \circ & & \circ & p \\
\bullet & & & \circ & & 6 & \circ & & & p \\
 & & & & & \bullet & 7 & & & \circ \\
 & & & & & p & \circ & 8 & & \circ \\
\bar{p} & p & p & p & \circ & p & \circ & & 9 & \circ \\
 & & p & & p & p & \circ & \bullet & \circ & 10
\end{array}\right) = \left(\begin{array}{cccccccccc}
1 & & & \bullet & & & & & & \\
\bullet & 2 & \bullet & \circ & & & & & d & d \\
 & & 3 & & \bullet & d & & & d & \\
 & & & 4 & & & \bullet & & & \\
 & \bullet & \circ & \circ & 5 & \circ & \bar{d} & & \circ & d \\
\bullet & & & \circ & & 6 & \circ & & & d \\
 & & & & & \bullet & 7 & & & \circ \\
 & & & & & d & \circ & 8 & & \circ \\
d & d & d & d & \circ & d & \circ & & 9 & \circ \\
 & & d & & d & d & \bar{d} & \bullet & \circ & 10
\end{array}\right)$$

FIG. 10. *The pruned filled matrix of the matrix in Figure* 1*, where $s$ denotes entries pruned by symmetric reduction; $\pi$ denotes entries pruned by partial path-symmetric reduction; $p$ denotes entries pruned by path-symmetric reduction; $d$ denotes entries pruned using the elimination dag, and a bar denotes entries not pruned by a lesser form of pruning.*

COROLLARY 4.2. *Let $A$ be upper BBT ordered. The row structure of $L_{r*}$ is given by*

$$\mathrm{Struct}(L_{r*}) = \{\, j \mid j \le r \textit{ and either } r \xmapsto{A} j \textit{ or } r \xmapsto{A} \xRightarrow{\bar{U}_{RR}} j \}.$$

*The row structure of $U_{r*}$ is given by*

$$\mathrm{Struct}(U_{r*}) = \left( \mathrm{Struct}(A_{r*}) \;\cup \bigcup_{k\,:\,\mathrm{FPNZ}(k)=r} \mathrm{Struct}(U_{k*}) \right) \setminus \{1, \ldots, r-1\}.$$

The expression for $\mathrm{Struct}(U_{r*})$ is the same as in the symmetric case [20, Theorem 8.1], where it is the key to efficient symbolic factorization.

**4.2.1. Pruning using the elimination tree.** Corollary 4.2 leads immediately to Algorithm ETREESYMBOLIC in Figure 11. The pruning strategy is based on the elimination tree and thus is equivalent to path-symmetric reduction as discussed in section 4.1.

However, there is another interpretation. Let $p = \mathrm{FPNZ}(r)$ be the parent of vertex $r$ in $T(A)$. By definition there is a cycle $p \xRightarrow{L} r \xRightarrow{U} p$. Since $A$ is upper BBT ordered, there is an edge $p \xmapsto{L} r$ (see [15, Theorem 6.5]). Thus there is a cycle $p \xmapsto{L} r \xRightarrow{U} p$; and path-symmetric reduction is the same as partial path-symmetric reduction. Algorithm ETREESYMBOLIC can be viewed as an implementation of either.

**4.2.2. Pruning using the elimination dags.** When the elimination dags are used in a symbolic factorization algorithm, they are usually constructed in parallel with the row structures of the factors [18, section 5.2]. That is, during the $r$th step,

```
Algorithm eTreeSymbolic
    Use Algorithm eTreeQ to construct the elimination tree
    for $r = 1$ to $n$ do
        $\text{Struct}(U_{r*}) = \text{Struct}(A_{r*}) \setminus \{1, \ldots, r-1\}$
        for each child $s$ of $r$ in $T(A)$ do
            $\text{Struct}(U_{r*}) = \text{Struct}(U_{r*}) \cup (\text{Struct}(U_{s*}) \setminus \{1, \ldots, r-1\})$
        end for
        $\text{Struct}(U^\theta_{r*}) = \text{Struct}(U_{r*}) \setminus \{\textsc{fpnz}(r)+1, \ldots, n\}$
        $\text{Struct}(L_{r*}) = \phi$
        for each vertex $i$ in $\text{Struct}(A_{r*}) \setminus \{1, \ldots, r-1\}$ do
            Add to $\text{Struct}(L_{r*})$ every vertex reachable from $i$
                through a path in $G(U^\theta_{RR})$
        end for
    end for
```

FIG. 11. *Symbolic factorization using the elimination tree (i.e., path-symmetric reduction) for an upper BBT ordered matrix.*

after computing $\text{Struct}(L_{r*})$ we use depth-first search to prune the row $L_{r*}$ to $L^o_{r*}$; and after computing $\text{Struct}(U_{r*})$ we use depth-first search to prune the column $U_{*r}$ to $U^o_{*r}$. (Pruning the row $U_{r*}$ to $U^o_{r*}$ would require the structures of higher-numbered rows of $U$.)

However, the situation is quite different when the matrix is upper BBT ordered. First, we need not construct the dag $G(L^o)$, which as noted earlier is the elimination tree with edges directed from parent to child [15, Theorem 6.3]. Second, we can use FPNZ$(r)$ to prune $U_{r*}$ to $U^\theta_{r*}$ as soon as we compute $\text{Struct}(U_{r*})$ so that there will be fewer edges to consider during transitive reduction.

Third, we can prune the *rows* of $U^\theta$ rather than the columns. Let vertex $s$ be a child of vertex $r$ in $T(A)$. Since $\ell_{rs}$ is the first off-diagonal nonzero in the $s$th column of $L$ (see [16, section 5.1]), we will not need $U^o_{s*}$ to compute the structure of any row of $L$ until we are ready to compute $\text{Struct}(L_{r*})$. Since the nonzero entries in $U^o_{s*}$ are in columns $s+1, \ldots, r = \textsc{fpnz}(s)$, pruning $U^\theta_{s*}$ to $U^o_{s*}$ requires only the corresponding rows of $U^o$. These rows will always be available if we transitively reduce the rows of $U^\theta$ corresponding to the children of $r$ in *descending* order when we are ready to compute the structure of $L_{r*}$.

Finally, we do not have to prune $U^\theta_{t*}$ to $U^o_{t*}$ if vertex $t$ is the last child of vertex $r$ in $T(A)$. By the definition of FPNZ$(t)$ we have $r \overset{L}{\Longrightarrow} t \overset{U}{\Longrightarrow} r$. Since the natural ordering of a BBT ordered matrix is a postordering and $t$ is the last child of $r$, we have $r = t + 1$. Thus the path $t \overset{U}{\Longrightarrow} r$ must be an edge in $U$; that is, $u_{tr} \neq 0$. Since $u_{tr}$ is the first nonzero in row $t$ of $U$, it must be retained in the elimination dag $G(U^o)$. On the other hand, since $r = \textsc{fpnz}(t)$, it is the only off-diagonal nonzero in row $t$ of $U^\theta$. Therefore $U^o_{t*} = U^\theta_{t*}$.

These observations lead to Algorithm EDAGSYMBOLIC in Figure 12.

**5. Experimental results.** In this section we demonstrate the efficiency of the algorithms to construct the elimination tree and to find a BBT postordering of an unsymmetric sparse matrix. We also compare the performance of Algorithms ETREESYMBOLIC and EDAGSYMBOLIC with existing codes for symbolic factorization. All of these algorithms were implemented in Fortran, and the experiments were performed on a dual 2.8 GHz Intel Xeon processor with 2 GB of RAM.

We started with 21 matrices from the University of Florida Sparse Matrix Collection [8] (see Table 2). Most are reducible, and for such matrices a block factorization (i.e., permute the rows and columns so that the matrix is in block upper triangular form and factor only the irreducible diagonal blocks) leads to less fill and work.

```
Algorithm edagSymbolic
  Use Algorithm eTreeQ to construct the elimination tree
  for $r = 1$ to $n$ do
    $\mathrm{Struct}(U_{r*}) = \mathrm{Struct}(A_{r*}) \setminus \{1, \ldots, r-1\}$
    for each child $s$ of $r$ in $T(A)$ do
      $\mathrm{Struct}(U_{r*}) = \mathrm{Struct}(U_{r*}) \cup (\mathrm{Struct}(U_{s*}) \setminus \{1, \ldots, r-1\})$
    end for
    $\mathrm{Struct}(U^{\theta}_{r*}) = \mathrm{Struct}(U_{r*}) \setminus \{\textsc{fpnz}(r)+1, \ldots, n\}$
    for each child $s$ of $r$ in $T(A)$, in descending order, do
      if $s$ is not the last child of $r$ then
        Prune $U^{\theta}_{s*}$ to obtain $U^{o}_{s*}$
    end for
    $\mathrm{Struct}(L_{r*}) = \phi$
    for each vertex $i$ in $\mathrm{Struct}(A_{r*}) \setminus \{1, \ldots, r-1\}$ do
      Add to $\mathrm{Struct}(L_{r*})$ every vertex reachable from $i$
        through a path in $G(U^{\theta}_{RR})$
    end for
  end for
```

FIG. 12. *Symbolic factorization using the elimination dags for an upper BBT ordered matrix.*

Thus we "preprocessed" each as follows:

1. Use the Dulmage–Mendelsohn decomposition [12] (`dmperm` in MATLAB) to identify the rows and columns in each irreducible diagonal block and delete all entries that lie in off-diagonal blocks.
2. Apply a random permutation symmetrically to the rows and columns to randomize the tie-breaking strategy in the next step.
3. Run the analysis phase of the multifrontal code `MA41_UNS` [3] (with `ICNTL(6) = 5`, `ICNTL(7) = 2`, and `ICNTL(8) = 7`) to find a transversal [10] (i.e., a column permutation that puts large entries on the diagonal) and a DMLS ordering [2] (i.e., a symmetric row and column permutation that preserves the large entries on the diagonal but reduces the off-diagonal fill).
4. Symmetrically permute the rows and columns to put the matrix in block diagonal form while preserving the (relative) DMLS ordering within each block.

We refer to the resulting matrix as $A$. Each problem was preprocessed eleven times (with different symmetric random permutations), and all numbers reported are medians over the corresponding set of eleven matrices.

Table 2 gives the basic statistics for the preprocessed problems. Here $n(A)$ is the size of $A$; $nz(A)/n(A)$ is the average number of nonzeros per row; $symm(A)$ is the percentage of nonzero off-diagonal elements $a_{ij}$ for which $a_{ij}$ and $a_{ji}$ are both nonzero; $fill(A)$ and $work(A)$ are the fill and work (in multiply-adds) when factoring $A$; and $t_s(A)$ is the time (in milliseconds) to perform a symbolic factorization of $A$ using symmetric reduction.

Table 3 gives the statistics for finding the elimination tree. Here $h(A)$ is the height of the elimination tree $T(A)$; $h(M)$ is the height of $T(M)$, the elimination tree for the symmetrized matrix $M = A + A^t$; $w_i(A)$ is the number of redundant edges in the set $W_i$, where an edge that can be pruned by both heuristics is included in both counts (see section 3.3); $t_e(A)$ is the time to find $T(A)$; $t_u(A)$ is the time to find $T(A)$ and an upper BBT reordering of that tree; and $t_p(A)$ is the time to perform a symbolic factorization of $A$ using path-symmetric reduction, *given* $T(A)$. Note the following:

- We must have $h(A) \leq h(M)$ since the parent of any node in $T(A)$ is an ancestor of that node in $T(M)$ (see [16, Theorem 7.1]).

TABLE 2
*Test problems.*

| Matrix | $n(A)$ | $\frac{nz(A)}{n(A)}$ | $symm(A)$ | $fill(A)$ | $work(A)$ | $t_s(A)$ (ms) |
|---|---|---|---|---|---|---|
| `Averous/epb1` | 14734 | 6.45 | 72.94 | $9.51\times10^5$ | $4.28\times10^7$ | 17.11 |
| `Bai/rw5151` | 5151 | 3.92 | .00 | $2.15\times10^5$ | $6.33\times10^6$ | 8.61 |
| `Goodwin/goodwin` | 7320 | 44.37 | 57.32 | $1.71\times10^6$ | $1.39\times10^8$ | 33.38 |
| `Graham/graham1` | 9035 | 34.27 | 59.88 | $1.38\times10^6$ | $1.01\times10^8$ | 25.38 |
| `Grund/bayer02` | 13935 | 4.05 | 7.57 | $2.79\times10^5$ | $3.96\times10^6$ | 6.24 |
| `Grund/bayer10` | 13436 | 4.84 | 6.12 | $2.82\times10^5$ | $4.41\times10^6$ | 6.08 |
| `HB/gemat11` | 4929 | 6.45 | 99.59 | $4.95\times10^4$ | $1.64\times10^5$ | 1.19 |
| `Hamrle/Hamrle2` | 5952 | 3.72 | 11.98 | $6.04\times10^4$ | $2.18\times10^5$ | 1.36 |
| `Hohn/fd12` | 7500 | 3.63 | 4.75 | $3.13\times10^5$ | $9.76\times10^6$ | 5.81 |
| `Hohn/fd15` | 11532 | 3.70 | 4.36 | $5.77\times10^5$ | $2.32\times10^7$ | 10.83 |
| `Hohn/fd18` | 16428 | 3.75 | 4.15 | $9.40\times10^5$ | $4.43\times10^7$ | 17.78 |
| `Hohn/sinc12` | 7500 | 35.81 | 25.40 | $7.02\times10^6$ | $2.82\times10^9$ | 195.25 |
| `Hollinger/g7jac040` | 11790 | 8.94 | 2.88 | $2.06\times10^6$ | $4.38\times10^8$ | 50.98 |
| `Lucifora/cell1` | 7055 | 4.26 | 79.63 | $1.86\times10^5$ | $2.93\times10^6$ | 4.35 |
| `Lucifora/cell2` | 7055 | 4.26 | 79.63 | $1.86\times10^5$ | $2.93\times10^6$ | 4.35 |
| `Nasa/barth` | 6691 | 3.12 | 15.55 | $8.35\times10^4$ | $8.89\times10^5$ | 2.03 |
| `Nasa/barth4` | 6019 | 3.83 | 16.98 | $1.52\times10^5$ | $2.61\times10^6$ | 3.61 |
| `Nasa/barth5` | 15606 | 3.65 | 14.44 | $3.13\times10^5$ | $4.16\times10^6$ | 9.24 |
| `Shen/e40r0100` | 17281 | 32.03 | 88.46 | $2.15\times10^6$ | $1.30\times10^8$ | 41.89 |
| `Shen/shermanACd` | 6136 | 6.89 | 99.57 | $1.12\times10^5$ | $2.00\times10^6$ | 2.46 |
| `TOKAMAK/utm5940` | 5940 | 14.02 | 54.80 | $7.36\times10^5$ | $4.28\times10^7$ | 13.35 |

TABLE 3
*Statistics for the elimination tree.*

| Matrix | $h(A)$ | $h(M)$ | $\frac{w_1(A)}{n(A)}$ | $\frac{w_2(A)}{n(A)}$ | $\frac{t_e(A)}{t_s(A)}$ | $\frac{t_u(A)}{t_s(A)}$ | $\frac{t_p(A)}{t_s(A)}$ |
|---|---|---|---|---|---|---|---|
| `Averous/epb1` | 1112 | 1122 | .13 | .06 | .48 | .56 | .97 |
| `Bai/rw5151` | 387 | 902 | .07 | .13 | 1.56 | 1.56 | .89 |
| `Goodwin/goodwin` | 1977 | 2062 | .45 | .34 | .48 | .49 | .92 |
| `Graham/graham1` | 1191 | 1338 | .34 | .22 | .65 | .67 | .97 |
| `Grund/bayer02` | 979 | 1291 | .14 | .08 | 1.29 | 1.45 | .96 |
| `Grund/bayer10` | 757 | 870 | .12 | .07 | 1.24 | 1.41 | .99 |
| `HB/gemat11` | 142 | 143 | .00 | .00 | 1.75 | 2.11 | 1.08 |
| `Hamrle/Hamrle2` | 645 | 813 | .30 | .28 | 2.12 | 2.42 | 1.07 |
| `Hohn/fd12` | 668 | 729 | .12 | .08 | .55 | .65 | .86 |
| `Hohn/fd15` | 986 | 1025 | .12 | .08 | .49 | .58 | .85 |
| `Hohn/fd18` | 1142 | 1179 | .12 | .08 | .46 | .54 | .86 |
| `Hohn/sinc12` | 3672 | 4332 | .79 | .67 | .09 | .09 | .96 |
| `Hollinger/g7jac040` | 1924 | 2295 | .59 | .58 | .74 | .78 | .90 |
| `Lucifora/cell1` | 720 | 778 | .20 | .11 | .70 | .84 | .92 |
| `Lucifora/cell2` | 720 | 778 | .20 | .11 | .70 | .84 | .91 |
| `Nasa/barth` | 258 | 294 | .22 | .22 | 1.47 | 1.74 | .94 |
| `Nasa/barth4` | 328 | 656 | .30 | .22 | 1.03 | 1.15 | .97 |
| `Nasa/barth5` | 427 | 976 | .31 | .29 | 1.75 | 1.86 | .98 |
| `Shen/e40r0100` | 1374 | 1384 | .20 | .16 | .67 | .72 | .95 |
| `Shen/shermanACd` | 289 | 289 | .01 | .01 | 1.12 | 1.33 | 1.07 |
| `TOKAMAK/utm5940` | 1109 | 1115 | .14 | .07 | .36 | .40 | .90 |
| Median | | | .20 | .11 | .70 | .84 | .95 |

Table 4
*Statistics for upper BBT ordered matrices.*

| Matrix | $\frac{\mathit{fill}(A_u)}{\mathit{fill}(A)}$ | $\frac{\mathit{work}(A_u)}{\mathit{work}(A)}$ | $\frac{t_s(A_u)}{t_s(A)}$ | $\frac{t_e(A_u)}{t_s(A_u)}$ | $\frac{t_p(A_u)}{t_s(A_u)}$ | $\frac{t_d(A_u)}{t_s(A_u)}$ |
|---|---|---|---|---|---|---|
| `Averous/epb1` | 1.07 | 1.01 | 1.13 | .39 | .94 | .89 |
| `Bai/rw5151` | 3.30 | 1.22 | 2.33 | .09 | .91 | 1.16 |
| `Goodwin/goodwin` | 1.07 | 1.06 | 1.08 | .42 | .93 | .90 |
| `Graham/graham1` | 1.21 | 1.32 | 1.23 | .49 | .94 | .92 |
| `Grund/bayer02` | 1.59 | 1.38 | 1.84 | .45 | .83 | 1.04 |
| `Grund/bayer10` | 1.44 | 1.25 | 1.63 | .54 | .89 | 1.04 |
| `HB/gemat11` | 1.00 | 1.00 | 1.03 | 1.68 | .96 | 1.13 |
| `Hamrle/Hamrle2` | 2.08 | 2.32 | 2.60 | .60 | .84 | 1.28 |
| `Hohn/fd12` | 1.49 | 1.48 | 1.81 | .25 | .85 | .83 |
| `Hohn/fd15` | 1.65 | 1.63 | 2.02 | .19 | .87 | .80 |
| `Hohn/fd18` | 1.62 | 1.61 | 2.09 | .17 | .86 | .79 |
| `Hohn/sinc12` | 1.08 | .99 | 1.32 | .06 | .97 | .98 |
| `Hollinger/g7jac040` | 2.08 | 1.59 | 2.69 | .06 | .83 | .93 |
| `Lucifora/cell1` | 2.32 | 1.85 | 2.00 | .30 | .92 | 1.07 |
| `Lucifora/cell2` | 2.32 | 1.85 | 1.97 | .31 | .93 | 1.08 |
| `Nasa/barth` | 1.95 | 2.00 | 2.20 | .50 | .87 | .93 |
| `Nasa/barth4` | 1.89 | 1.24 | 2.32 | .27 | .89 | .93 |
| `Nasa/barth5` | 2.36 | 1.69 | 2.62 | .25 | .90 | .98 |
| `Shen/e40r0100` | 1.05 | 1.07 | 1.10 | .61 | .92 | .88 |
| `Shen/shermanACd` | 1.02 | 1.01 | 1.02 | 1.05 | .97 | 1.04 |
| `TOKAMAK/utm5940` | 1.08 | 1.09 | 1.09 | .32 | .93 | .88 |
| Median | 1.59 | 1.32 | 1.84 | .32 | .91 | .93 |

- The number of edges in the sets $W_1$ and $W_2$ is small, most likely because of the pruning implicit in working with the quotient graph.
- The time to find $T(A)$ is never appreciably larger than the time for symbolic factorization, and the time to find an upper BBT ordering as well is only slightly larger.
- As expected, path-symmetric reduction is faster than symmetric reduction. However, the improvement is less than the cost of finding $T(A)$. Thus symmetric reduction remain the preferred approach unless the elimination tree is needed for other reasons.

Table 4 gives the fill and work for the upper BBT reordered matrix $A_u$ of $A$ and the symbolic factorization times $t_s(A_u)$, $t_p(A_u)$, and $t_d(A_u)$ using symmetric reduction, Algorithm eTreeSymbolic, and Algorithm edagSymbolic, respectively. Note the following:

- BBT reordering generally increases both the fill and the work (by medians of 59% and 32%, respectively), which offsets some of the advantages of BBT ordered matrices [16].
- Again path-symmetric reduction (Algorithm eTreeSymbolic) is faster than symmetric reduction; but while the improvement is usually less than the cost of finding $T(A_u)$, that should not be necessary in this case.
- Algorithm edagSymbolic is faster than symmetric reduction more often than not; but again the improvement is usually less than the cost of finding $T(A_u)$.

Thus path-symmetric reduction is the preferred approach.

$$A = \begin{pmatrix} 1 & \bullet & & & & & \bullet & \bullet & \bullet & & & \\ & 2 & \bullet & & & & \bullet & \bullet & \bullet & & & \\ & & 3 & \bullet & & & \bullet & \bullet & \bullet & & & \\ & & & 4 & \bullet & & \bullet & \bullet & \bullet & & & \\ & & & & 5 & \bullet & \bullet & \bullet & \bullet & & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & 6 & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & & & & & \bullet & 7 & \bullet & \circ & \circ & \circ & \circ \\ & & & & & \bullet & \circ & 8 & \bullet & \circ & \circ & \circ \\ & & & & & \bullet & \circ & \circ & 9 & \bullet & \circ & \circ \\ \bullet & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & 10 & \bullet & \circ \\ \bullet & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & 11 & \bullet \\ \bullet & \circ & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & 12 \end{pmatrix}$$

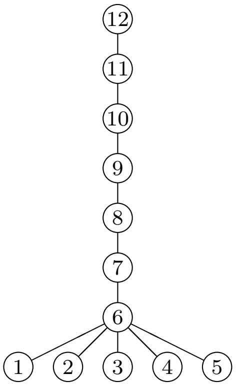


Fig. 13. *A contrived matrix and its elimination tree.*

Table 5
*Times for the* 1000 × 1000 *version of the matrix in Figure* 13.

| Algorithm | Time (ms) |
|---|---|
| Algorithm eTreeQ | 5.62 |
| Symbolic factorization by | |
| Symmetric reduction | 82.34 |
| Path-symmetric reduction | 8.14 |
| Algorithm eTreeSymbolic | 7.60 |
| Algorithm edagSymbolic | 7.46 |

The upper BBT matrix in Figure 13 demonstrates the advantage that the algorithms considered here can have over symmetric reduction. Table 5 gives the times for a larger version of that matrix.

## 6. Concluding remarks.

In this paper we have presented algorithms for constructing the elimination tree and finding a BBT postordering of an unsymmetric matrix. Experiments indicate that the run-times of these algorithms are comparable with those for symbolic factorization. We have also shown how the elimination tree and BBT postorderings can be used to improve the performance of a symbolic factorization algorithm. However, for the problems considered here path-symmetric reduction, the best of the new algorithms, is only better than symmetric reduction if the elimination tree is available.

These ideas also have been applied with great effect in the unsymmetric multifrontal method [16]. Here we suggest two other applications.

### 6.1. Unsymmetric supernodes.

By grouping together adjacent rows and/or columns with the same nonzero structure in $A^+$ and treating them as a dense matrix for storage and computation, we can improve the efficiency of symmetric [5] and unsymmetric [9] sparse numerical factorization. These groupings of row/column indices are called *supernodes*.

There are several ways to define an unsymmetric supernode [9], including the following:

- A $T_1$ supernode is a range[8] $r:s$ of columns of $L$ and rows of $U$, such that the diagonal block $A^+(r:s, r:s)$ is full and each row of $L(s+1:n, r:s)$ and each column of $U(r:s, s+1:n)$ is either full or zero.

---

[8] We use the MATLAB notations $p:q$, denoting the sequence $p, p+1, \ldots, q$, and $M(P,Q)$, denoting the submatrix of $M$ defined by the row sequence $P$ and the column sequence $Q$.

- A $T_3$ supernode is a range $r\!:\!s$ of columns of $L$ and $U$, such that the diagonal block $A^+(r\!:\!s, r\!:\!s)$ is full and each row of $L(s{+}1\!:\!n, r\!:\!s)$ is either full or zero.

Since the diagonal block $A^+(r\!:\!s, r\!:\!s)$ is dense, necessary (but not sufficient) conditions for $r\!:\!s$ to be a $T_1$ or $T_3$ supernode are that the nodes $r, \ldots, s$ lie on an upward path in the elimination tree $T(A)$ and that only node $r$ has more than one child. This simplifies the identification of supernodes and suggests how to relax the definitions to increase their number and/or sizes (see [4, 11] for the symmetric case). Moreover, since the nodes in a supernode are numbered sequentially during a BBT ordering, such an ordering may also reveal more or larger supernodes.

**6.2. Diagonal Markowitz ordering.** Suppose that we want to minimize the work required to factor an unsymmetric matrix by symmetrically reordering the rows and columns (i.e., the same permutation is applied to both rows and columns). A greedy algorithm selects at each stage the pivot $i$ in the reduced matrix for which the product $r_i c_i$ is smallest. This is the diagonal Markowitz ordering [2].

To implement this algorithm we need an efficient scheme to maintain the nonzero structure of the reduced matrix (i.e., the connectivity in the Schur complement). One approach is to use quotient graphs [21, 2]. The techniques used to speed construction of the elimination tree in section 3 seem to have immediate application to making the maintenance of these quotient graphs more efficient.

**Acknowledgments.** We thank Patrick Amestoy, who provided us with a version of `MA41_UNS` that includes the DMLS ordering; and the referees, whose insightful comments helped us to improve the presentation.

## REFERENCES

[1] A. V. Aho, M. R. Garey, and J. D. Ullman, *The transitive reduction of a directed graph*, SIAM J. Comput., 1 (1972), pp. 131–137.

[2] P. R. Amestoy, X. S. Li, and E. G. Ng, *Diagonal Markowitz scheme with local symmetrization*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 228–244.

[3] P. R. Amestoy and C. Puglisi, *An unsymmetrized multifrontal LU factorization*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 553–569.

[4] C. Ashcraft and R. Grimes, *The influence of relaxed supernode partitions on the multifrontal method*, ACM Trans. Math. Software, 15 (1989), pp. 291–309.

[5] C. Ashcraft, R. Grimes, J. Lewis, B. Peyton, and H. Simon, *Progress in sparse matrix methods for large sparse linear systems on vector supercomputers*, Internat. J. Supercomputer Appl., 1 (1987), pp. 10–30.

[6] A. Chang, *Application of sparse matrix methods in electric power system analysis*, in Sparse Matrix Proceedings, R. A. Willoughby, ed., Report RA1 (#11707), IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1968, pp. 113–122.

[7] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1992.

[8] T. A. Davis, *University of Florida Sparse Matrix Collection*; also available online at http://www.cise.ufl.edu/research/sparse/matrices/.

[9] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu, *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 720–755.

[10] I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.

[11] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[12] A. L. Dulmage and N. S. Mendelsohn, *Coverings of bipartite graphs*, Canad. J. Math., 10 (1958), pp. 517–534.

[13] S. C. Eisenstat and J. W. H. Liu, *Exploiting structural symmetry in unsymmetric sparse symbolic factorization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 202–211.

[14] S. C. Eisenstat and J. W. H. Liu, *Structural representations of Schur complements in sparse matrices*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer-Verlag, New York, 1993, pp. 85–100.

[15] S. C. Eisenstat and J. W. H. Liu, *The theory of elimination trees for sparse unsymmetric matrices*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 686–705.

[16] S. C. Eisenstat and J. W. H. Liu, *A tree-based dataflow model for the unsymmetric multifrontal method*, Electron. Trans. Numer. Anal., 21 (2005), pp. 1–19.

[17] H. N. Gabow, *Path-based depth-first search for strong and biconnected components*, Inform. Process. Lett., 74 (2000), pp. 107–114.

[18] J. R. Gilbert and J. W. H. Liu, *Elimination structures for unsymmetric sparse LU factors*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 334–352.

[19] A. Gupta, *Improved symbolic and numerical factorization algorithms for unsymmetric sparse matrices*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 529–552.

[20] J. W. H. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.

[21] G. Pagallo and C. Maulino, *A bipartite quotient graph model for unsymmetric matrices*, in Numerical Methods, V. Pereyra and A. Reinoza, eds., Lecture Notes in Math. 1005, Springer-Verlag, New York, 1983, pp. 227–239.

[22] D. J. Rose and R. E. Tarjan, *Algorithmic aspects of vertex elimination on directed graphs*, SIAM J. Appl. Math., 34 (1978), pp. 176–197.

[23] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., 1 (1972), pp. 146–160.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1382–1388

# ON THE CONVERGENCE OF GENERAL STATIONARY LINEAR ITERATIVE METHODS FOR SINGULAR LINEAR SYSTEMS*

ZHI-HAO CAO†

**Abstract.** Recently, Lee et al. have published an interesting paper [*SIAM J. Matrix Anal. Appl.*, 28 (2006), pp. 634–641] concerning the energy norm convergence of general stationary linear iterative methods for semidefinite linear systems. In this paper, we first consider the convergence of general stationary linear iterative methods for general singular consistent linear systems and show that the convergence and the quotient convergence are equivalent. Then we consider the convergence of general stationary iterative methods for the semidefinite systems and clarify some issues in Lee et al.'s paper.

**Key words.** iterative methods, singular linear systems, matrix splitting, convergence, quotient convergence, energy norm convergence

**AMS subject classifications.** 65F10, 65F15

**DOI.** 10.1137/060671243

## 1. Introduction.

We consider the problem of finding a solution $x \in \mathcal{R}^n$ to

$$Ax = b, \tag{1.1}$$

where $A \in \mathcal{R}^{n,n}$ is a given singular matrix and $b \in \mathcal{R}^n$ is a given vector in the range of $A$. A classical stationary linear iterative method to solve (1.1) can be obtained by using a splitting [2, 3, 7, 12] of $A$: $A = M - N$,

$$Mx^{(k)} = Nx^{(k-1)} + b, \quad k = 1, 2, \dots,$$

which is equivalent to the following iterative scheme:

$$x^{(k)} = x^{(k-1)} - M^{-1}(Ax^{(k-1)} - b), \quad k = 1, 2, \dots. \tag{1.2}$$

The matrix $T = I - M^{-1}A \equiv M^{-1}N$ determined by the splitting $A = M - N$ is called the iteration matrix and the splitting is called induced by the iteration matrix $T$ [7]. When $A$ is nonsingular, an iteration matrix $T$ induces an unique splitting of $A$. Thus, constructing a classical stationary iterative scheme is equivalent to making a splitting of $A$. However, when $A$ is singular, there are infinitely many splittings of $A$ induced by the same iteration matrix $T$ (cf. [2, 5]). Therefore, we extend the classical stationary linear iterative method to the general stationary linear iterative method by replacement of $M^{-1}$ in (1.2) with $M^\dagger$ in a straightforward manner

$$x^{(k)} = x^{(k-1)} - M^\dagger(Ax^{(k-1)} - b), \quad k = 1, 2, \dots, \tag{1.3}$$

where $M^\dagger$ is the Moore–Penrose generalized inverse of $M$ (see [1]). The iteration matrix is

$$T = I - M^\dagger A. \tag{1.4}$$

---

*Received by the editors October 2, 2006; accepted for publication (in revised form) by R. Nabben September 14, 2007; published electronically January 4, 2008. This work was supported by NSFC project 10471027.

http://www.siam.org/journals/simax/29-4/67124.html

†School of Mathematical Sciences and Laboratory of Mathematics for Nonlinear Sciences, Fudan University, Shanghai 200433, People's Republic of China (zcao@fudan.edu.cn; zhcao@cableplus.com.cn).

We emphasize that (1.3) is not proposed as a computational scheme but rather as useful for theoretical analysis.

Definition 1.1. *We say that the iterative scheme* (1.3) *is convergent if, for any initial guess* $x^{(0)} \in \mathcal{R}^n$*, the iteration sequence* $\{x^{(k)}\}$ *produced by the iterative scheme converges to a solution* $x_*$ *of* (1.1) *as* $k \to \infty$.

Let $A^\dagger$ be the Moore–Penrose inverse of $A$, and then $P \equiv A^\dagger A$ is an orthogonal projection onto $\mathcal{N}(A)^\perp$, the orthogonal complement of the null space $\mathcal{N}(A)$. Then $x_{**} \equiv A^\dagger b$ is the unique solution of (1.1) with the least 2-norm (see [1, 3]).

Definition 1.2. *We say that the iterative scheme* (1.3) *is quotient convergent if, for any initial guess* $x^{(0)} \in \mathcal{R}^n$*, the iteration sequence* $\{x^{(k)}\}$ *produced by the iterative scheme is such that the sequence* $\{Px^{(k)}\}$ *converges to the least* 2*-norm solution* $x_{**}$ *of* (1.1) *as* $k \to \infty$.

**2. General singular systems.** For any matrix $B \in \mathcal{R}^{n,n}$ let $\gamma(B)$ denote the pseudospectral radius of $B$, i.e.

$$\gamma(B) = \max\{|\lambda| : \lambda \in \sigma(B) \setminus \{1\}\}, \tag{2.1}$$

where $\sigma(B)$ is the set of the eigenvalues of $B$.

We now consider the convergence of the iterative method (1.3). Let $e^{(k)} = x^{(k)} - x_{**}$ be the iteration error of $x^{(k)}$, and then from (1.3) we have

$$e^{(k)} = (I - M^\dagger A)e^{(k-1)} = Te^{(k-1)} = T^k e^{(0)}. \tag{2.2}$$

Thus, the fact that $e^{(k)} \to x_* - x_{**} \in \mathcal{N}(A)(k \to \infty)$; i.e., the iterative scheme (1.3) is convergent if and only if $T$ is semiconvergent [3]; i.e.,

$$T^k \to T_\infty(k \to \infty), \tag{2.3}$$

and $T_\infty$ is a projection onto the null space $\mathcal{N}(A)$.

It is well known (see [10]) that a matrix $T$ is semiconvergent if and only if there is a nonsingular matrix $S$ such that

$$S^{-1}TS = \begin{pmatrix} I & \\ & \widetilde{T} \end{pmatrix}, \tag{2.4}$$

where the spectral radius $\rho(\widetilde{T})$ of $\widetilde{T}$ satisfies $\rho(\widetilde{T}) < 1$, i.e., $\widetilde{T}$ is a convergent matrix. Therefore, $T$ is semiconvergent if and only if the following two conditions are satisfied:

(a) The pseudospectral radius $\gamma(T) \equiv \rho(\widetilde{T}) < 1$.

(b) $\operatorname{rank}(I - T) = \operatorname{rank}((I - T)^2)$ or, equivalently, $index_1(T) \equiv index(I - T) = 1$ (cf. [3] for the definitions of $index_\lambda(B)$ and $index(B)$ for a square matrix $B$).

From (1.4) and (2.4) we have

$$M^\dagger A = S \begin{pmatrix} 0 & \\ & I - \widetilde{T} \end{pmatrix} S^{-1} \tag{2.5}$$

and

$$T_\infty = S \begin{pmatrix} I & \\ & 0 \end{pmatrix} S^{-1}. \tag{2.6}$$

Let $S = [S_1, S_2]$, (2.5) and (2.6) imply that $T_\infty$ is a projection, and

$$Span(S_1) = \mathcal{R}(T_\infty) = \mathcal{N}(M^\dagger A), \tag{2.7}$$

$$Span(S_2) = \mathcal{N}(T_\infty) = \mathcal{R}(M^\dagger A). \tag{2.8}$$

From (2.7) we know that $T_\infty$ is a projection onto $\mathcal{N}(M^\dagger A)$. Therefore, the condition

$$\mathcal{N}(M^\dagger A) = \mathcal{N}(A) \tag{2.9}$$

is a necessary and sufficient condition for the projection $T_\infty$ being a projection onto $\mathcal{N}(A)$.

From the discussion above we can obtain the following convergence theorem (see [9]).

THEOREM 2.1. *The general stationary linear iterative scheme*

$$x^{(k)} = x^{(k-1)} - M^\dagger(Ax^{(k-1)} - b), \quad k = 1, 2, \ldots,$$

*is convergent if and only if the following three conditions are fulfilled:*

(a) $\gamma(T) < 1$, *where* $T \equiv I - M^\dagger A$ *is the iteration matrix,*

(b) $index_1(T) \equiv index(I - T) = 1$,

(c) $\mathcal{N}(M^\dagger A) = \mathcal{N}(A)$.

We now consider the relationship between the convergence and the quotient convergence. We obtain the following theorem, which extends the corresponding result in [4], and it is the one of the main results in this paper.

THEOREM 2.2. *For the general stationary linear iterative scheme* (1.3)

$$x^{(k)} = x^{(k-1)} - M^\dagger(Ax^{(k-1)} - b), \quad k = 1, 2, \ldots,$$

*the convergence and the quotient convergence are equivalent.*

*Proof.* We need only to show that the quotient convergence implies the convergence, since the latter is obviously implies the former.

The iterative scheme can be written as

$$x^{(k)} = x^{(k-1)} - M^\dagger A(x^{(k-1)} - x_{**}), \quad k = 1, 2, \ldots .$$

If $\mathcal{N}(M^\dagger A) \neq \mathcal{N}(A)$, then there exists an initial guess $x^{(0)}$ such that $e^{(0)} \equiv x^{(0)} - x_{**} \in \mathcal{N}(M^\dagger A)$ and $e^{(0)} \notin \mathcal{N}(A)$. With this $x^{(0)}$, the iterative scheme (1.3) produces an iteration sequence $\{x^{(k)}\} : x^{(k)} = x^{(0)}, k = 1, 2, \ldots$. Obviously, $Px^{(0)} - x_{**} \equiv Pe^{(0)} \neq 0$, since $A(Pe^{(0)}) = Ae^{(0)} \neq 0$. Therefore, the quotient convergence implies

$$\mathcal{N}(M^\dagger A) = \mathcal{N}(A). \tag{2.10}$$

From (2.2) we have

$$Pe^{(k)} = PT^k e^{(0)}.$$

Thus, the iterative scheme (1.3) is quotient convergent if and only if

$$PT^k \to 0 (k \to \infty), \tag{2.11}$$

since $PT^k e^{(0)} \to 0 (k \to \infty)$ for all $e^{(0)} \in \mathcal{R}^n$. Since $P \equiv A^\dagger A$ is an orthogonal projection onto $\mathcal{N}(A)^\perp$, there exists an orthogonal matrix $\widehat{S}$ such that

$$\widehat{S}^T P \widehat{S} = \begin{pmatrix} 0 & \\ & I \end{pmatrix}. \tag{2.12}$$

Let $\widehat{S} = [\widehat{S}_1, \widehat{S}_2]$, where (using (2.10))

$$Span(\widehat{S}_1) = \mathcal{N}(A) = \mathcal{N}(M^\dagger A) \quad \text{and} \quad Span(\widehat{S}_2) = \mathcal{N}(A)^\perp = \mathcal{N}(M^\dagger A)^\perp. \tag{2.13}$$

Since $T = I - M^\dagger A$, we have $T\widehat{S}_1 = \widehat{S}_1$. Thus, the matrix $\widehat{S}^T T \widehat{S}$ can be expressed as the following form:

$$\widehat{S}^T T \widehat{S} = \begin{pmatrix} I & R_{12} \\ & R_{22} \end{pmatrix}.$$

If $index_1(T) \equiv index(I - T) > 1$, then $1 \in \sigma(R_{22})$. In this case, we have

$$PT^k = \widehat{S} \begin{pmatrix} 0 & \\ & R_{22}^k \end{pmatrix} \widehat{S}^T \not\to 0 (k \to \infty).$$

This means that $index_1(T) \equiv index(I - T) = 1$ is the necessary condition for the quotient convergence.

Since $index_1(T) = 1$, there exists a nonsingular matrix $S$ such that

$$T = S \begin{pmatrix} I & \\ & \widetilde{T} \end{pmatrix} S^{-1}, \quad \text{and hence} \quad M^\dagger A = S \begin{pmatrix} 0 & \\ & I - \widetilde{T} \end{pmatrix} S^{-1}, \tag{2.14}$$

where $1 \notin \sigma(\widetilde{T})$ and $\rho(\widetilde{T}) = \gamma(T)$. Let $S = [S_1, S_2]$, where (using (2.10))

$$Span(S_1) = \mathcal{N}(M^\dagger A) = \mathcal{N}(A) \quad \text{and} \quad Span(S_2) = \mathcal{R}(M^\dagger A). \tag{2.15}$$

Then the matrix $S = [S_1, S_2]$ can be expressed as

$$[S_1, S_2] = [\widehat{S}_1, \widehat{S}_2] \begin{pmatrix} G_{11} & G_{12} \\ & G_{22} \end{pmatrix}, \tag{2.16}$$

where $G_{11}$ and $G_{22}$ are nonsingular. From (2.16) we have

$$S^{-1} = \begin{pmatrix} G_{11}^{-1} & \widehat{G}_{12} \\ & G_{22}^{-1} \end{pmatrix} \widehat{S}^T. \tag{2.17}$$

From (2.12), (2.14), (2.16), and (2.17) we have

$$\begin{aligned} PT^k &= \widehat{S} \begin{pmatrix} 0 & \\ & I \end{pmatrix} \widehat{S}^T S \begin{pmatrix} I & \\ & \widetilde{T}^k \end{pmatrix} S^{-1} \\ &= [0, \widehat{S}_2] \begin{pmatrix} G_{11} & G_{12} \\ & G_{22} \end{pmatrix} \begin{pmatrix} I & \\ & \widetilde{T}^k \end{pmatrix} \begin{pmatrix} G_{11}^{-1} & \widehat{G}_{12} \\ & G_{22}^{-1} \end{pmatrix} \widehat{S}^T \\ &= [0, \widehat{S}_2 G_{22} \widetilde{T}^k G_{22}^{-1}] \widehat{S}^T. \end{aligned}$$

Thus, the fact that $PT^k \to 0 (k \to \infty)$ implies $\widetilde{T}^k \to 0 (k \to \infty)$, which is equivalent to $\gamma(T) = \rho(\widetilde{T}) < 1$.

In summary, we have showed that the quotient convergence of the iterative scheme (1.3) implies that all three conditions (a), (b), and (c) in Theorem 2.1 are satisfied. Thus, the quotient convergence implies the convergence. ☐

By applying Theorem 2.2 we obtain the following result.

THEOREM 2.3. *The general stationary linear iterative scheme*

$$x^{(k)} = x^{(k-1)} - M^\dagger (A x^{(k-1)} - b), \quad k = 1, 2, \ldots,$$

*is convergent if and only if*

$$r^{(k)} \equiv A x^{(k)} - b \to 0 (k \to \infty) \quad \forall x^{(0)} \in \mathcal{R}^n. \tag{2.18}$$

*Proof.* Equation (2.18) is equivalent to

$$Ae^{(k)} \to 0 (k \to \infty) \quad \forall e^{(0)} \in \mathcal{R}^n, \tag{2.19}$$

since $e^{(k)} = x^{(k)} - x_{**}$. (2.19) implies

$$Pe^{(k)} \to 0 (k \to \infty) \quad \forall e^{(0)} \in \mathcal{R}^n, \tag{2.20}$$

since $P = A^{\dagger}A$. Inversely, (2.20) implies (2.19), since $AP = AA^{\dagger}A = A$. Thus, (2.19) and (2.20) are equivalent to each other. Now (2.20) is equivalent to the quotient convergence. Applying Theorem 2.2 (2.18) is also equivalent to the convergence. ☐

Theorem 2.3 shows that (2.18) can be used as the definition of convergence (cf. Definition 1.1) of the general stationary iterative scheme (1.3).

## 3. Semidefinite systems.

Now we consider the symmetric positive semidefinite systems; i.e., we assume the matrix $A \in \mathcal{R}^{n,n}$ in (1.1) is symmetric positive semidefinite. Lee et al. [8] call the iterative scheme (1.3) being energy norm convergent if and only if

$$|I - M^{\dagger}A|_A < 1. \tag{3.1}$$

Obviously, (3.1) implies (2.18). Thus, from Theorem 2.3 the energy norm convergence implies the convergence.

For symmetric positive definite systems and the classical stationary linear iterative scheme, i.e., in the case that $A$ is symmetric positive definite and $M$ is invertible, we have the following well known theorem (see [11, 6]).

THEOREM 3.1. *For symmetric positive definite systems $Ax = b$, the classical stationary linear iterative scheme*

$$x^{(k)} = x^{(k-1)} - M^{-1}(Ax^{(k-1} - b), \quad k = 1, 2, \dots,$$

*is energy norm convergent if and only if $M^T + M - A$ is symmetric positive definite.*

This convergence result can easily be extended to positive semidefinite systems.

THEOREM 3.2. *For symmetric positive semidefinite systems $Ax = b$, the classical stationary linear iterative scheme*

$$x^{(k)} = x^{(k-1)} - M^{-1}(Ax^{(k-1} - b), \quad k = 1, 2, \dots,$$

*is energy norm convergent if and only if $M^T + M - A$ is symmetric positive definite on $\mathcal{R}(M^{-1}A)$.*

*Proof.* It is easy to see that (cf. [5])

$$A - (I - M^{-1}A)^T A(I - M^{-1}A) = (M^{-1}A)^T(M^T + M - A)(M^{-1}A). \tag{3.2}$$

From (3.2) we immediately obtain the conclusion of the theorem. ☐

Lee et al. [8] have further extended these convergence results to the case of the general stationary linear iterative scheme and obtain the following necessary and sufficient conditions for the energy norm convergence (see Theorem 4.4 in [8]):

(A1) $\mathcal{R}(A) \subset \mathcal{R}(M)$ or, equivalently, $\mathcal{N}(M^T) \subset \mathcal{N}(A)$,
(A2) $M^T + M - A$ is symmetric positive definite on $\mathcal{R}(M^{\dagger}A)$,

or, equivalently,

(A1) $\mathcal{R}(A) \subset \mathcal{R}(M)$ or, equivalently, $\mathcal{N}(M^T) \subset \mathcal{N}(A)$,
(A2a) there exists $\omega \in (0, 1)$ such that $(M^{\dagger}Ax, M^{\dagger}Ax)_A \le \omega(M^{\dagger}Ax, Ax)$ for all $x \in \mathcal{R}^n$,
(A2b) there exists $\alpha > 0$ such that $(M^{\dagger}Ax, M^{\dagger}Ax)_A \ge \alpha(M^{\dagger}Ax, M^{\dagger}Ax)$ for all $x \in \mathcal{R}^n$.

However, this result has some problems to clarify.

First we compare the condition (A1) with the condition $\mathcal{N}(M^\dagger A) = \mathcal{N}(A)$, which is necessary for the convergence.

THEOREM 3.3. *Let* $A \in \mathcal{R}^{n,n}$ *be a symmetric matrix, and* $M \in \mathcal{R}^{n,n}$*. Then the condition* (A1) *implies the condition* $\mathcal{N}(M^\dagger A) = \mathcal{N}(A)$*.*

*Proof.* It is easy to see that the condition $\mathcal{N}(M^\dagger A) = \mathcal{N}(A)$ is equivalent to the condition $\mathcal{N}(M^\dagger) \bigcap \mathcal{R}(A) = \{0\}$.

Now if $\mathcal{N}(M^T) \subset \mathcal{N}(A)$, then $\mathcal{N}(M^T) \bigcap \mathcal{R}(A) = \{0\}$, since $A$ is symmetric. Noting that $\mathcal{N}(M^\dagger) = \mathcal{N}(M^T)$ the conclusion of the theorem follows. ☐

Now we give a semidefinite system and a general stationary linear iterative scheme which is energy norm convergent, but the conditions (A1) and (A2) are not satisfied.

*Example.* We take the semidefinite matrix $A$ and the vector $b \in \mathcal{R}(A)$ as

$$A = \begin{pmatrix} 0 & \\ & 1 \end{pmatrix} \quad \text{and} \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

respectively. The matrix $M$ is

$$M = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix},$$

and hence, the matrix $M^\dagger$ is

$$\begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix}.$$

Obviously, the least 2-norm solution of $Ax = b$ is $x_{**} = \binom{0}{1}$, and the iteration matrix $T$ is

$$T = I - M^\dagger A = \begin{pmatrix} 1 & \\ & 0 \end{pmatrix}.$$

Since

$$(I - M^\dagger A)^T A (I - M^\dagger A) = T^T A T = 0,$$

$|I - M^\dagger A|_A = 0$. Thus, the general stationary linear iterative scheme is energy norm convergent. Now we have

$$\mathcal{N}(M^T) = Span\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}\right\} \quad \text{and} \quad \mathcal{N}(A) = Span\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right\},$$

and, thus, the condition (A1) is not satisfied. We note that the condition $\mathcal{N}(M^\dagger A) = \mathcal{N}(A)$ is satisfied, since $M^\dagger A = \left(\begin{smallmatrix} 0 & \\ & 1 \end{smallmatrix}\right) = A$.

Now we consider the condition (A2). We have

$$M^T + M - A = \frac{1}{2}\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix},$$

and

$$(M^\dagger A)^T (M^T + M - A)(M^\dagger A) = 0.$$

Thus, condition (A2) is also not satisfied.

Finally, we point out that the key problem in Theorem 4.4 of [8] is that (A1) is not a necessary condition for energy norm convergence (cf. Theorem 3.3). It is easy to show that if the condition (A1) is assumed to be satisfied, then the condition (A2) is sufficient and necessary for energy norm convergence. Therefore, Theorem 4.4 of [8] can be revised or correctly formulated as follows.

THEOREM 3.4. *Let* (A1) *be satisfied. Then the iterative scheme* (1.3) *is energy norm convergent if and only if* (A2) *is satisfied or, equivalently, if and only if* (A2a) *and* (A2b) *are satisfied.*

*Proof.* By (A1) and the fact that $MM^\dagger M = M$ it is easy to see that (cf. Lemma 4.1 in [8])

$$MM^\dagger A = A, \tag{3.3}$$

and we have showed that (A1) implies (cf. Theorem 3.3)

$$\mathcal{N}(M^\dagger A) = \mathcal{N}(A). \tag{3.4}$$

By using (3.3) we have (cf. (3.2))

$$\begin{aligned} A - (I - M^\dagger A)^T A(I - M^\dagger A) &= (M^\dagger A)^T A + AM^\dagger A - (M^\dagger A)^T A(M^\dagger A) \\ &= (M^\dagger A)^T M(M^\dagger A) + A(M^\dagger)^T M^T(M^\dagger A) - (M^\dagger A)^T A(M^\dagger A) \\ &= (M^\dagger A)^T (M + M^T - A)(M^\dagger A). \end{aligned} \tag{3.5}$$

From (3.5) and (3.4) we obtain that the iterative scheme (1.3) is energy norm convergent if and only if (A2) is satisfied.

By Lemma 4.3 in [8] (A2) is equivalent to (A2a) and (A2b). ☐

We note that Theorem 3.4 is still an extension of Theorem 3.2 to general stationary linear iterative schemes for positive semidefinite linear systems.

**Acknowledgments.** I gratefully acknowledge the reviewer and Prof. R. Nabben for many comments and suggestions that helped improve this paper greatly.

## REFERENCES

[1] A. BEN-ISRAEL AND T. N. E. GREVILLE, *Generalized Inverses, Theory and Application*, Wiley, New York, 1974.

[2] M. BENZI AND D. B. SZYLD, *Existence and uniqueness of splittings for stationary iterative methods with applications to alternating methods*, Numer. Math., 76 (1997), pp. 309–321.

[3] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, 2nd ed., SIAM, Philadelphia, 1994.

[4] Z.-H. CAO, *On the convergence of iterative methods for solving singular linear systems*, J. Comput. Appl. Math., 145 (2002), pp. 1–9.

[5] Z.-H. CAO, *A note on properties of splittings of singular symmetric positive semidefinite matrices*, Numer. Math., 88 (2001), pp. 603–606.

[6] Z.-H. CAO, *A note on P-regular splittings of Hermitian matrix*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1392–1393.

[7] P. J. LANZKRON, D. J. ROSE, AND D. B. SZYLD, *Convergence of nested classical iterative methods for linear systems*, Numer. Math., 58 (1991), pp. 685–702.

[8] Y.-J. LEE, J.-B. WU, J. XU, AND L. ZIKATANOV, *On the convergence of iterative methods for semidefinite linear systems*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 634–641.

[9] G. I. MARCHUK AND Y. KUZNETZOV, *Iterative Methods and Quadratic Functionals* (in Russian), Science Press, Norvosibirsk, 1972.

[10] C. D. MEYER, JR., AND R. J. PLEMMONS, *Convergent powers of a matrix with applications to iterative methods for singular linear systems*, SIAM J. Numer. Anal., 14 (1977), pp. 699–705.

[11] D. M. YOUNG, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.

[12] R. S. VARGA, *Matrix Iterative Analysis*, 2nd ed., Springer–Verlag, Berlin, Heidelberg, 2000.

SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1389–1410

# STRUCTURED MAPPING PROBLEMS FOR MATRICES ASSOCIATED WITH SCALAR PRODUCTS. PART I: LIE AND JORDAN ALGEBRAS*

D. STEVEN MACKEY†, NILOUFER MACKEY†, AND FRANÇOISE TISSEUR‡

**Abstract.** Given a class of structured matrices $\mathbb{S}$, we identify pairs of vectors $x, b$ for which there exists a matrix $A \in \mathbb{S}$ such that $Ax = b$, and we also characterize the set of all matrices $A \in \mathbb{S}$ mapping $x$ to $b$. The structured classes we consider are the Lie and Jordan algebras associated with orthosymmetric scalar products. These include (skew-)symmetric, (skew-)Hamiltonian, pseudo(skew-)Hermitian, persymmetric, and perskew-symmetric matrices. Structured mappings with extremal properties are also investigated. In particular, structured mappings of minimal rank are identified and shown to be unique when rank one is achieved. The structured mapping of minimal Frobenius norm is always unique, and explicit formulas for it and its norm are obtained. Finally the set of all structured mappings of minimal 2-norm is characterized. Our results generalize and unify existing work, answer a number of open questions, and provide useful tools for structured backward error investigations.

**Key words.** Lie algebra, Jordan algebra, scalar product, bilinear form, sesquilinear form, orthosymmetric, adjoint, structured matrix, backward error, Hamiltonian, skew-Hamiltonian, Hermitian, complex symmetric, skew-symmetric, persymmetric, perskew-symmetric, minimal rank, minimal Frobenius norm, minimal 2-norm

**AMS subject classifications.** 15A04, 15A57, 15A60, 15A63, 65F30, 65F35

**DOI.** 10.1137/060657856

## 1. Introduction.

The problem of finding all matrices $A$ that map a given nonzero vector $x \in \mathbb{K}^n$ to a given vector $b \in \mathbb{K}^m$, where $\mathbb{K}$ is a fixed field, can be solved using elementary means [10]. Trenkler [21] recently revisited this problem, giving a solution using generalized inverses:

$$A = bx^{\dagger} + Z(I_n - xx^{\dagger}), \tag{1.1}$$

where $I_n$ is the $n \times n$ identity matrix, $Z \in \mathbb{K}^{m\times n}$ is arbitrary, and $x^{\dagger}$ is any generalized inverse of $x$. In this work we restrict the permissible transformations to a class of structured matrices $\mathbb{S} \subset \mathbb{K}^{n\times n}$ and consider the following *structured mapping problems.*

EXISTENCE. *For which vectors $x, b$ does there exist some $A \in \mathbb{S}$ such that $Ax = b$?*

CHARACTERIZATION. *Determine the set $\mathcal{S} = \{\, A \in \mathbb{S} : \ Ax = b \,\}$ of all structured mappings taking $x$ to $b$.*

We present a complete, unified solution for these two problems when $\mathbb{S}$ is the Lie or Jordan algebra associated with an orthosymmetric scalar product. These $\mathbb{S}$ include, for example, symmetric and skew-symmetric, Hermitian, pseudo-Hermitian and skew-Hermitian, Hamiltonian, persymmetric, and perskew-symmetric matrices.

---

[*] Received by the editors April 21, 2006; accepted for publication (in revised form) by R. Nabben June 21, 2007; published electronically January 9, 2008.
http://www.siam.org/journals/simax/29-4/65785.html

[†] Department of Mathematics, Western Michigan University, Kalamazoo, MI 49008 (steve.mackey@wmich.edu, nil.mackey@wmich.edu, http://homepages.wmich.edu/~mackey/).

[‡] School of Mathematics, The University of Manchester, Sackville Street, Manchester, M60 1QD, UK (ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/). This author's work was supported by Engineering and Physical Sciences Research Council grant GR/S31693.

We will assume that $x \neq 0$ throughout, since both problems have trivial solutions if $x = 0$.

Answers to some particular instances of these structured mapping problems can be found in the literature. Liu and Leake [9, Lem. 1] show that for $x, b \in \mathbb{R}^n$, $x$ can be mapped to $b$ by a real skew-symmetric matrix if and only if $x$ and $b$ are orthogonal. Khatri and Mitra [8] and later Sun [17] address the existence and characterization problems for the matrix equation $AX = B$, where $X$, $B$ are matrices and the unknown $A$ is Hermitian; the skew-Hermitian and complex symmetric cases are covered in [19]. Restricting the results of [8], [17], and [19] to the case when $X$ and $B$ are vectors yields one among the many representations of the set $\mathcal{S}$ identified in this paper. Structured mapping problems for double structures, for structures that do not arise in the context of a scalar product, and for some specific nonlinear structures have also been investigated (see [5], [6], [15], [18], and [22] for examples).

One of our motivations for studying these problems stems from the analysis of structured backward errors in the solutions to structured linear systems and structured eigenproblems [7], [19], [20]. Recall that a backward error of an approximate solution $y$ to a linear system $Ax = b$ is a measure of the smallest perturbation $E$ such that $(A + E)y = b$. When $A$ is in some linearly structured class $\mathbb{S}$ one may want to require $E$ to have the same structure; the structured backward error is then a measure of the smallest structured perturbation $E$ such that $Ey = r := b - Ay$. Hence solving the structured mapping problem is the first step towards obtaining explicit expressions for structured backward errors.

For any linear matrix structure $\mathbb{S}$ it is possible to obtain a characterization of the structured mapping set $\mathcal{S}$ using the Kronecker product approach of [4], which we briefly outline here. The equation $Ax = b$ is first rewritten as $(x^T \otimes I_n)\operatorname{vec}(A) = b$, where $\otimes$ denotes the Kronecker product and vec is the operator that stacks the columns of a matrix into one long vector. The linear nature of the matrix structure is then encoded by $\operatorname{vec}(A) = \Pi_{\mathbb{S}} p$, where $\Pi_{\mathbb{S}}$ is an $n^2 \times m$ pattern matrix giving (in essence) a basis for the structured class $\mathbb{S}$, and $p$ is an $m$-dimensional vector of parameters ($m = \dim \mathbb{S} \leq n^2$). Hence

$$\mathcal{S} = \{\, A \in \mathbb{K}^{n \times n} : \ (x^T \otimes I_n)\Pi_{\mathbb{S}} p = b, \ \operatorname{vec}(A) = \Pi_{\mathbb{S}} p \,\}. \tag{1.2}$$

Note that there may be no solution to the system $(x^T \otimes I_n)\Pi_{\mathbb{S}} p = b$ if $(x^T \otimes I_n)\Pi_{\mathbb{S}}$ is rank deficient or if the system is overdetermined ($n > m$). When they exist, solutions can be obtained from the singular value decomposition of $(x^T \otimes I_n)\Pi_{\mathbb{S}}$. In particular, if the system is underdetermined and consistent, and if the pattern matrix $\Pi_{\mathbb{S}}$ is chosen so that $\|p\|_2 = \|A\|_F$ for all $A \in \mathbb{S}$ (i.e., $\Pi_{\mathbb{S}}$ contains an orthonormal basis for $\mathbb{S}$ in the Frobenius inner product), then the solution $A \in \mathbb{S}$ with minimal Frobenius norm is given in terms of the pseudoinverse by $p = \big((x^T \otimes I_n)\Pi_{\mathbb{S}}\big)^{+} b$. As a result a computable expression for the structured backward error is obtained:

$$\eta_F(y) = \min\{\, \|E\|_F : (A + E)y = b, \ E \in \mathbb{S} \,\} = \big\|\big((y^T \otimes I_n)\Pi_{\mathbb{S}}\big)^{+}(b - Ay)\big\|_2. \tag{1.3}$$

There are several disadvantages associated with this Kronecker product approach. The existence of structured solutions to $Ax = b$ may not be easy to check. In addition, the set $\mathcal{S}$ of all structured mappings is given only implicitly by (1.2). Also, among all solutions in $\mathcal{S}$, it is difficult to distinguish ones with special properties, other than that of minimal Frobenius norm. The structured backward error expression in (1.3) is expensive to evaluate and difficult to compare with its unstructured counterpart $\|b - Ay\|_2$.

By contrast, the approach presented in this paper gives easy-to-check conditions for the existence problem and an *explicit* solution for the characterization problem when $\mathbb{S}$ is the Lie or Jordan algebra of a scalar product. The set $\mathcal{S}$ is rewritten as

$$\mathcal{S} = B + \{\, A \in \mathbb{S} : \ Ax = 0 \,\}, \tag{1.4}$$

where $B$ is any particular solution of the nonhomogeneous mapping problem. We provide a set of possible particular solutions $B$ for a given class $\mathbb{S}$ and given vectors $x$ and $b$, thus giving multiple ways of representing $\mathcal{S}$. This enables one to more easily identify structured mappings with minimal rank or minimal Frobenius norm and to readily derive bounds for the ratio between the structured and unstructured backward errors. A multiplicative representation, by contrast with the additive representation in (1.4), is used to characterize the set of all minimal 2-norm structured mappings in $\mathcal{S}$. From this characterization, minimal 2-norm mappings of minimal rank and minimal 2-norm mappings of minimal Frobenius norm can be identified.

To give an idea of the scope of the paper, we give here an illustration of what is obtained by applying our general results to a particular structure $\mathbb{S}$, in this case the Lie algebra of complex skew-symmetric matrices. For given $x, b \in \mathbb{C}^n$ our results imply that

$$\mathcal{S} := \{\, A \in \mathbb{C}^{n\times n} : \ Ax = b, \ A^T = -A \,\} \ \text{is nonempty} \quad \Longleftrightarrow \quad x^T b = 0\,,$$

and that

$$\mathcal{S} \ = \ \{\, bw^T - wb^T + (I - vx^T)L(I - xv^T) : \ L \in \mathbb{C}^{n\times n}, \ L^T = -L \,\}\,, \tag{1.5}$$

where $w, v \in \mathbb{C}^n$ are any fixed but arbitrary vectors chosen such that $w^T x = v^T x = 1$. All mappings in $\mathcal{S}$ of the form $bw^T - wb^T$ (corresponding to setting $L = 0$ in (1.5)) have minimal rank two, and the choice $w = \bar{x}/\|x\|_2^2$, $L = 0$ gives the unique mapping $A_{\text{opt}}$ of minimal Frobenius norm:

$$A_{\text{opt}} = (b\bar{x}^T - \bar{x}b^T)/\|x\|_2^2\,, \quad \|A_{\text{opt}}\|_F = \min_{A\in\mathcal{S}} \|A\|_F = \sqrt{2}\|b\|_2/\|x\|_2\,. \tag{1.6}$$

The set $\mathcal{M} := \{\ A \in \mathcal{S} : \ \|A\|_2 = \min_{B\in\mathcal{S}} \|B\|_2 \ \}$ of all minimal 2-norm mappings can be characterized by

$$\mathcal{M} = \frac{\|b\|_2}{\|x\|_2}\left\{ U^T \operatorname{diag}\left( \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right], S \right) U : \ S \in \mathbb{C}^{(n-2)\times(n-2)}, \ S^T = -S, \ \|S\|_2 \le 1 \right\},$$

where $U^*[\,x\ \bar{b}\,] = [\,\|x\|_2 e_1\ \ \|b\|_2 e_2\,]$; i.e., $U$ is the unitary factor of the QR factorization of $[\,x\ \bar{b}\,]$ with $R$ forced to have positive entries. For this structure $\mathbb{S}$ it turns out that $A_{\text{opt}} \in \mathcal{M}$, so $A_{\text{opt}}$ is simultaneously a mapping of minimal rank, minimal Frobenius norm, and minimal 2-norm. As a consequence of (1.6) an explicit formula for the structured backward error in (1.3) for this class $\mathbb{S}$ is given for the Frobenius norm by

$$\eta_F(y) = \sqrt{2}\,\frac{\|Ay - b\|_2}{\|y\|_2}\,,$$

which is immediately seen to differ from its unstructured counterpart by a factor of only $\sqrt{2}$. For the 2-norm the structured and unstructured backward errors are equal.

In summary, the results here generalize and unify existing work, answer a number of open questions, and provide useful tools for the investigation of structured backward errors. After some preliminaries in section 2, a complete solution to the existence and characterization problems is presented in sections 3 and 4. In section 5 we identify structured mappings of minimal rank, minimal Frobenius norm, and minimal 2-norm, and investigate their uniqueness. Some technical proofs are given in the appendix.

## 2. Preliminaries.

**2.1. Scalar products.** A *bilinear form* on $\mathbb{K}^n$ ($\mathbb{K} = \mathbb{R}, \mathbb{C}$) is a map $(x, y) \mapsto \langle x, y\rangle$ from $\mathbb{K}^n \times \mathbb{K}^n$ to $\mathbb{K}$, which is linear in each argument. If $\mathbb{K} = \mathbb{C}$, the map $(x, y) \mapsto \langle x, y\rangle$ is a *sesquilinear form* if it is conjugate linear in the first argument and linear in the second. To a bilinear form on $\mathbb{K}^n$ is associated a unique $M \in \mathbb{K}^{n\times n}$ such that $\langle x, y\rangle = x^T My$ for all $x, y \in \mathbb{K}^n$; if the form is sesquilinear, $\langle x, y\rangle = x^*My$ for all $x, y \in \mathbb{C}^n$, where the superscript $*$ denotes the conjugate transpose. The form is said to be *nondegenerate* when $M$ is nonsingular.

A bilinear form is symmetric if $\langle x, y\rangle = \langle y, x\rangle$ or, equivalently, if $M^T = M$, and skew-symmetric if $\langle x, y\rangle = -\langle y, x\rangle$ or, equivalently, if $M^T = -M$. A sesquilinear form is Hermitian if $\langle x, y\rangle = \overline{\langle y, x\rangle}$ and skew-Hermitian if $\langle x, y\rangle = -\overline{\langle y, x\rangle}$. The matrices associated with such forms are Hermitian and skew-Hermitian, respectively.

We will use the term *scalar product* to mean a nondegenerate bilinear or sesquilinear form on $\mathbb{K}^n$. When we have more than one scalar product under consideration, we will denote $\langle x, y\rangle$ by $\langle x, y\rangle_{\mathrm{M}}$, using the matrix $M$ defining the form as a subscript to distinguish the forms under discussion.

**2.2. Adjoints.** The *adjoint* of $A$ with respect to the scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$, denoted by $A^\star$, is uniquely defined by the property $\langle Ax, y\rangle_{\mathrm{M}} = \langle x, A^\star y\rangle_{\mathrm{M}}$ for all $x, y \in \mathbb{K}^n$. It can be shown that the adjoint is given explicitly by

$$A^\star = \begin{cases} M^{-1}A^TM & \text{for bilinear forms,} \\ M^{-1}A^*M & \text{for sesquilinear forms.} \end{cases}$$

The following properties of adjoint, all analogous to properties of transpose (or conjugate transpose), follow easily and hold for all scalar products.

LEMMA 2.1. *$(A + B)^\star = A^\star + B^\star$, $(AB)^\star = B^\star A^\star$, $(A^{-1})^\star = (A^\star)^{-1}$ and*

$$(\alpha A)^\star = \begin{cases} \alpha A^\star & \textit{for bilinear forms,} \\ \overline{\alpha} A^\star & \textit{for sesquilinear forms.} \end{cases}$$

The involutory property $(A^\star)^\star = A$ does not hold for all scalar products; this issue is discussed in section 2.4.

**2.3. Lie and Jordan algebras.** Associated with $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is a Lie algebra $\mathbb{L}$ and a Jordan algebra $\mathbb{J}$, defined by

$$\mathbb{L} := \left\{A \in \mathbb{K}^{n\times n} : \langle Ax, y\rangle_{\mathrm{M}} = -\langle x, Ay\rangle_{\mathrm{M}}\ \forall x, y \in \mathbb{K}^n\right\} = \left\{A \in \mathbb{K}^{n\times n} : A^\star = -A\right\},$$

$$\mathbb{J} := \left\{A \in \mathbb{K}^{n\times n} : \langle Ax, y\rangle_{\mathrm{M}} = \langle x, Ay\rangle_{\mathrm{M}}\ \forall x, y \in \mathbb{K}^n\right\} = \left\{A \in \mathbb{K}^{n\times n} : A^\star = A\right\}.$$

All the structured matrices considered in this paper belong to one of these two classes. Note that $\mathbb{L}$ and $\mathbb{J}$ are linear subspaces of $\mathbb{K}^{n\times n}$. Table 2.1 shows a sample of well-known structured matrices in some $\mathbb{L}$ or $\mathbb{J}$ associated with a scalar product.

**2.4. Orthosymmetric and unitary scalar products.** Scalar products for which vector orthogonality is a symmetric relation, i.e.,

$$\langle x, y\rangle_{\mathrm{M}} = 0 \Leftrightarrow \langle y, x\rangle_{\mathrm{M}} = 0\ \forall x, y \in \mathbb{K}^n,$$

will be referred to as *orthosymmetric scalar products* [12], [13]. One can show that $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is orthosymmetric if and only if it satisfies any one (and hence all) of the following equivalent properties:

Table 2.1

*Structured matrices associated with some orthosymmetric scalar products.*

$$R = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix},\ J = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix},\ \Sigma_{p,q} = \begin{bmatrix} I_p & 0 \\ 0 & -I_q \end{bmatrix} \text{ with } p+q=n.$$

| Space | $M$ | Adjoint<br>$A^\star$ | Jordan algebra<br>$\mathbb{J} = \{A : A^\star = A\}$ | Lie algebra<br>$\mathbb{L} = \{A : A^\star = -A\}$ |
|---|---|---|---|---|
| Symmetric bilinear forms | | | | |
| $\mathbb{R}^n$ | $I$ | $A^T$ | Symmetrics | Skew-symmetrics |
| $\mathbb{C}^n$ | $I$ | $A^T$ | Complex symmetrics | Complex skew-symmetrics |
| $\mathbb{R}^n$ | $\Sigma_{p,q}$ | $\Sigma_{p,q}A^T\Sigma_{p,q}$ | Pseudosymmetrics | Pseudoskew-symmetrics |
| $\mathbb{C}^n$ | $\Sigma_{p,q}$ | $\Sigma_{p,q}A^T\Sigma_{p,q}$ | Complex pseudosymm. | Complex pseudoskew-symm. |
| $\mathbb{R}^n$ | $R$ | $RA^TR$ | Persymmetrics | Perskew-symmetrics |
| Skew-symmetric bilinear forms | | | | |
| $\mathbb{R}^{2n}$ | $J$ | $-JA^TJ$ | Skew-Hamiltonians | Hamiltonians |
| $\mathbb{C}^{2n}$ | $J$ | $-JA^TJ$ | Complex $J$-skew-symm. | Complex $J$-symmetrics |
| Hermitian sesquilinear forms | | | | |
| $\mathbb{C}^n$ | $I$ | $A^*$ | Hermitian | Skew-Hermitian |
| $\mathbb{C}^n$ | $\Sigma_{p,q}$ | $\Sigma_{p,q}A^*\Sigma_{p,q}$ | Pseudo-Hermitian | Pseudoskew-Hermitian |
| Skew-Hermitian sesquilinear forms | | | | |
| $\mathbb{C}^{2n}$ | $J$ | $-JA^*J$ | $J$-skew-Hermitian | $J$-Hermitian |

1. The adjoint with respect to $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is involutory, i.e., $(A^\star)^\star = A$ for all $A \in \mathbb{K}^{n\times n}$.
2. $M = \alpha M^T$ with $\alpha = \pm 1$ for bilinear forms; $M = \alpha M^*$ with $\alpha \in \mathbb{C}$, $|\alpha| = 1$ for sesquilinear forms.
3. $\mathbb{K}^{n\times n} = \mathbb{L} \oplus \mathbb{J}$.

See [12, Thm. A.4] or [13, Thm. 1.6] for a proof of this equivalence along with a list of additional equivalent properties. The second property says that orthosymmetric bilinear forms are always either symmetric or skew-symmetric. On the other hand, an orthosymmetric sesquilinear form $\langle x, y\rangle_{\mathrm{M}} = x^*My$, where $M = \alpha M^*$, $|\alpha| = 1$, $\alpha \in \mathbb{C}$, is always closely tied to a Hermitian form: defining the Hermitian matrix $H = \bar{\alpha}^{1/2} M$ gives $\langle x, y\rangle_{\mathrm{H}} = \bar{\alpha}^{1/2} \langle x, y\rangle_{\mathrm{M}}$ for all $x, y \in \mathbb{C}^n$. Consequently, the Jordan algebra of $\langle\cdot,\cdot\rangle_{\mathrm{H}}$ is identical to the Jordan algebra of $\langle\cdot,\cdot\rangle_{\mathrm{M}}$:

$$\langle Ax, y\rangle_{\mathrm{H}} = \langle x, Ay\rangle_{\mathrm{H}} \quad\Leftrightarrow\quad \bar{\alpha}^{1/2} \langle Ax, y\rangle_{\mathrm{M}} = \bar{\alpha}^{1/2} \langle x, Ay\rangle_{\mathrm{M}} \quad\Leftrightarrow\quad \langle Ax, y\rangle_{\mathrm{M}} = \langle x, Ay\rangle_{\mathrm{M}} \ .$$

Similarly, the Lie algebras of $\langle\cdot,\cdot\rangle_{\mathrm{H}}$ and $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ are also identical. Thus a result established for Hermitian sesquilinear forms immediately translates into a corresponding result for orthosymmetric sesquilinear forms. Up to a scalar multiple, then, there are really only three distinct types of orthosymmetric scalar products: symmetric bilinear, skew-symmetric bilinear, and Hermitian sesquilinear. We will, however, continue to include separately stated results (without separate proofs) for skew-Hermitian forms for convenience, as this is a commonly occurring special case.

The results in this paper hold only for orthosymmetric scalar products, which as just mentioned are those for which the useful and simplifying property $(A^\star)^\star = A$

holds for all matrices [12], [13]. For these scalar products, adjoints of rank-one matrices will often be needed,

$$
\begin{aligned}
&(yz^TM)^\star = zy^TM^T && \text{for orthosymmetric bilinear forms,}\\
&(yz^*M)^\star = zy^*M^* && \text{for orthosymmetric sesquilinear forms.}
\end{aligned}
\tag{2.1}
$$

Some of our results will require the extra property that the scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is also *unitary*, that is, $\beta M$ is unitary for some $\beta > 0$ [12]. One can show that in unitary scalar products, "the stars commute," i.e., $(A^*)^\star = (A^\star)^*$ for all $A \in \mathbb{K}^{n\times n}$ and that for every unitarily invariant norm $\|\cdot\|$, $\|A^\star\| = \|A\|$ for all $A \in \mathbb{K}^{n\times n}$ [13, Thm. 1.8]. Finally, note that important classes of structured matrices arise in the context of scalar products that are both orthosymmetric and unitary, as witnessed by the entries in Table 2.1 (for all of which $\alpha = \pm 1$ and $\beta = 1$). The results in this paper are not confined to just the examples in the table, however.

### 2.5. Projections.

Projections that map $x$ to the zero vector form a key part in our solution to the structured mapping problems.

Since the matrix $M$ of the scalar product is nonsingular, given a nonzero $x \in \mathbb{K}^n$ one can always construct many $w \in \mathbb{K}^n$ such that $\langle w, x\rangle_{\mathrm{M}} = 1$. For example, when $x$ is nonisotropic (i.e., $\langle x, x\rangle_{\mathrm{M}} \neq 0$), $w = x/\langle x, x\rangle_{\mathrm{M}}$ will work for bilinear forms, and $w = x/\overline{\langle x, x\rangle_{\mathrm{M}}}$ can be used for sesquilinear forms. If $x$ is isotropic (i.e., $\langle x, x\rangle_{\mathrm{M}} = 0$), choose $k$ so that $x_k \neq 0$; then $w = M^{-T}e_k/x_k$ will have the desired property for bilinear forms, and $w = M^{-*}e_k/\overline{x}_k$ will work for sesquilinear forms.

With $w$ chosen so that $\langle w, x\rangle_{\mathrm{M}} = 1$, it is easy to show that for bilinear forms, $xw^TM$ is idempotent and hence a projection with range span$\{x\}$. Replacing $^T$ by $^*$ gives a similar result for sesquilinear forms. The complementary projections $P_w$ defined by

$$
P_w := \begin{cases} I - xw^TM, & \langle w, x\rangle_{\mathrm{M}} = 1 \quad \text{for bilinear forms,}\\ I - xw^*M, & \langle w, x\rangle_{\mathrm{M}} = 1 \quad \text{for sesquilinear forms}\end{cases}
\tag{2.2}
$$

have kernel span$\{x\}$, and in particular map $x$ to the zero vector.

## 3. The existence problem.

Throughout the rest of the paper we assume that $x, b \in \mathbb{K}^n$ with $x \neq 0$, but that any $b$ is allowed unless otherwise stated. For a scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ we will denote by $\mathbb{S}$ the corresponding Jordan algebra $\mathbb{J}$ or Lie algebra $\mathbb{L}$ and will write

$$
\mathcal{S} := \{\, A \in \mathbb{S} : \ Ax = b \,\},\ \mathcal{J} := \{\, A \in \mathbb{J} : \ Ax = b \,\},\ \mathcal{L} = \{\, A \in \mathbb{L} : \ Ax = b \,\}
\tag{3.1}
$$

for the associated structured mapping sets. Note that $\mathcal{S} = \mathcal{J}$ when $\mathbb{S} = \mathbb{J}$ and $\mathcal{S} = \mathcal{L}$ when $\mathbb{S} = \mathbb{L}$.

As a preliminary step towards solving the existence problem, we show that the projections given in (2.2) can be used to construct maps that send $x$ to $b$.

LEMMA 3.1. *Let $x \neq 0$, and let $w \in \mathbb{K}^n$ be chosen so that $\langle w, x\rangle_{\mathrm{M}} = 1$ with $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ orthosymmetric. Then ${}^{\pm}B_w x = b$, where ${}^{\pm}B_w$ is defined by*

$$
{}^{\pm}B_w := \begin{cases} bw^TM \pm (bw^TM)^\star P_w & \text{for bilinear forms,}\\ bw^*M \pm (bw^*M)^\star P_w & \text{for sesquilinear forms.}\end{cases}
\tag{3.2}
$$

*Note that ${}^{+}B_w$ and ${}^{-}B_w$ have rank at most two.*

*Proof.* Since $P_w x = 0$ and $\langle w, x\rangle_{\mathrm{M}} = 1$, we immediately conclude ${}^{\pm}B_w x = b$. Next, by (2.1) we see that in the bilinear case, $(bw^T M)^{\star} P_w = wb^T M^T P_w$, which is a rank-one matrix, and hence ${}^{+}B_w$, ${}^{-}B_w$ are the sum of two matrices of rank one. The proof in the sesquilinear case is similar. ☐

Thus, since ${}^{+}B_w$, ${}^{-}B_w$ are always solutions to the unstructured mapping problem, they should be consistent with (1.1), which captures all solutions. Now $\langle w, x\rangle_{\mathrm{M}} = 1$ implies $w^T M x = 1$ in the bilinear case. Since any row vector $u^T$ with the property $u^T x = 1$ is a generalized inverse $x^{\dagger}$ for the map $x : \mathbb{R} \to \mathbb{R}^n$, we can take $x^{\dagger}$ to be $w^T M$. Rewriting (3.2) for the bilinear case we get

$$
{}^{\pm}B_w = bx^{\dagger} \pm wb^T M^T (I - xx^{\dagger}), \tag{3.3}
$$

which is of the form given by Trenkler in (1.1) with $Z = \pm wb^T M^T$. The argument for the sesquilinear case is similar, with the role of $x^{\dagger}$ being played by $w^* M$. It is worth observing that once the parameter $w^T$ is chosen, both $x^{\dagger}$ and $Z$ in (3.3) are determined, and thus we are confining our attention to a constrained subset of the maps given by (1.1).

We still have to determine when a structured mapping exists, and what role ${}^{+}B_w$, ${}^{-}B_w$ play in such a mapping. The next theorem characterizes pairs of vectors $x$, $b$ for which there exists $A \in \mathbb{L}$ or $A \in \mathbb{J}$ such that $Ax = b$. When a structured mapping exists, we show that either ${}^{-}B_w$ or ${}^{+}B_w$ will be in the Lie or Jordan algebra, thus yielding a constructive proof of existence.

THEOREM 3.2 (existence for $\mathbb{L}$, $\mathbb{J}$). *Let $\langle \cdot, \cdot\rangle_{\mathrm{M}}$ be an orthosymmetric scalar product. Then for any given pair of vectors $x, b \in \mathbb{K}^n$ with $x \neq 0$ and associated structured mapping sets $\mathcal{J}$ and $\mathcal{L}$ in* (3.1),

$$
\mathcal{J} \neq \emptyset \iff \langle b, x\rangle_{\mathrm{M}} = \langle x, b\rangle_{\mathrm{M}}, \tag{3.4}
$$

$$
\mathcal{L} \neq \emptyset \iff \langle b, x\rangle_{\mathrm{M}} = -\langle x, b\rangle_{\mathrm{M}}. \tag{3.5}
$$

*In particular, when $\langle b, x\rangle_{\mathrm{M}} = \langle x, b\rangle_{\mathrm{M}}$ then ${}^{+}B_w \in \mathcal{J}$, and when $\langle b, x\rangle_{\mathrm{M}} = -\langle x, b\rangle_{\mathrm{M}}$, ${}^{-}B_w \in \mathcal{L}$.*

*Proof.* ($\Rightarrow$) Since $Ax = b$, in all cases we have

$$
A \in \mathbb{J} \Rightarrow \langle b, x\rangle_{\mathrm{M}} = \langle Ax, x\rangle_{\mathrm{M}} = \langle x, Ax\rangle_{\mathrm{M}} = \langle x, b\rangle_{\mathrm{M}},
$$

$$
A \in \mathbb{L} \Rightarrow \langle b, x\rangle_{\mathrm{M}} = \langle Ax, x\rangle_{\mathrm{M}} = \langle x, -Ax\rangle_{\mathrm{M}} = -\langle x, b\rangle_{\mathrm{M}}.
$$

($\Leftarrow$) By Lemma 3.1 we know that ${}^{+}B_w$, ${}^{-}B_w$ as defined in (3.2) map $x$ to $b$. It suffices to prove that when $\langle b, x\rangle_{M} = \langle x, b\rangle_{M}$, ${}^{+}B_w \in \mathbb{J}$, and when $\langle b, x\rangle_{M} = -\langle x, b\rangle_{M}$, ${}^{-}B_w \in \mathbb{L}$. Using Lemma 2.1 and the expressions for the adjoints given in (2.1), we have for bilinear forms

$$
\begin{aligned}
{}^{\pm}B_w &= bw^T M \pm (bw^T M)^{\star}(I - xw^T M) \\
&= bw^T M \pm (bw^T M)^{\star} \mp w(b^T M^T x) w^T M \\
&= bw^T M \pm (bw^T M)^{\star} \mp \langle x, b\rangle_{\mathrm{M}} ww^T M,
\end{aligned}
$$

and on the other hand,

$$
\begin{aligned}
{}^{\pm}B_w^{\star} &= (bw^T M)^{\star} \pm (I - wx^T M^T) bw^T M \\
&= (bw^T M)^{\star} \pm bw^T M \mp w(x^T M^T b) w^T M \\
&= (bw^T M)^{\star} \pm bw^T M \mp \langle b, x\rangle_{\mathrm{M}} ww^T M.
\end{aligned}
$$

If $\langle b, x\rangle_M = \langle x, b\rangle_M$, then clearly ${}^+B_w = {}^+B_w^\star$ so that ${}^+B_w \in \mathbb{J}$. If $\langle b, x\rangle_M = -\langle x, b\rangle_M$, then ${}^-B_w^\star = (bw^TM)^\star - bw^TM - \langle x, b\rangle_{\mathrm{M}} ww^TM = -{}^-B_w$ so that ${}^-B_w \in \mathbb{L}$. The proof in the sesquilinear case is similar. □

The existence conditions in (3.4)–(3.5) are made more explicit in Corollary 3.3 for the main types of orthosymmetric scalar products. Observe that sometimes a condition on $\langle b, x\rangle_{\mathrm{M}}$ is needed, while in other cases a structured mapping exists with no restrictions at all on $x$ and $b$.

COROLLARY 3.3. *Let* $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ *be any orthosymmetric scalar product for which*

$$M = \begin{cases} \pm M^T & \text{for bilinear forms,} \\ \pm M^* & \text{for sesquilinear forms.} \end{cases}$$

*Then for any given pair of vectors* $x,\ b \in \mathbb{K}^n$ *with* $x \neq 0$*, let* $\mathcal{S}$ *denote either* $\mathcal{J}$ *or* $\mathcal{L}$ *as in* (3.1)*. Then* $\mathcal{S} \neq \emptyset$ *if and only if the conditions given in the following table hold:*

| *Scalar product* | $\mathcal{J} \neq \emptyset$ | $\mathcal{L} \neq \emptyset$ |
|---|---|---|
| *Symmetric bilinear* | *always* | $\langle b, x\rangle_{\mathrm{M}} = 0$ |
| *Skew-symmetric bilinear* | $\langle b, x\rangle_{\mathrm{M}} = 0$ | *always* |
| *Hermitian sesquilinear* | $\langle b, x\rangle_{\mathrm{M}} \in \mathbb{R}$ | $\langle b, x\rangle_{\mathrm{M}} \in i\mathbb{R}$ |
| *Skew-Hermitian sesquilinear* | $\langle b, x\rangle_{\mathrm{M}} \in i\mathbb{R}$ | $\langle b, x\rangle_{\mathrm{M}} \in \mathbb{R}$ |

*Proof.* The conditions in the table follow from Theorem 3.2 and the definitions of symmetric and skew-symmetric bilinear forms and of Hermitian and skew-Hermitian sesquilinear forms. □

Theorem 3.2 and Corollary 3.3 unify and generalize existence results in [9] for real skew-symmetric matrices, in [8] and [17] for symmetric and Hermitian matrices, in [19] for complex symmetric and skew-Hermitian structures, and in [16, Lem. 5.1] for real persymmetric matrices, which are particular instances of Lie and Jordan algebras associated with different bilinear and sesquilinear forms on $\mathbb{R}^n$ and $\mathbb{C}^n$ (see Table 2.1).

## 4. The characterization problem.

We turn now to the task of determining the set of *all* matrices that map $x$ to $b$ and belong to a Lie or Jordan algebra.

LEMMA 4.1. *Let* $\mathbb{S}$ *denote the Lie or Jordan algebra of any orthosymmetric scalar product. Then*

(a) $A \in \mathbb{S} \Rightarrow Q^\star A Q \in \mathbb{S}$ *for all* $Q$*; that is,* $\star$*-congruence preserves* $\mathbb{L}$ *and* $\mathbb{J}$ *structures;*

(b) $\{P_w^\star S P_w : S \in \mathbb{S}\} \subseteq \{A \in \mathbb{S} : Ax = 0\}$*, where* $P_w$ *is any particular one of the projection matrices defined in* (2.2)*;*

(c) *for any* $w \in \mathbb{K}^n$ *such that* $\langle w, x\rangle_M = 1$, $A \in \mathbb{S}$, $Ax = 0 \Longrightarrow A = P_w^\star A\, P_w$.

*Proof.* (a) This is a direct consequence of adjoint being involutory in orthosymmetric scalar products.

(b) Follows immediately from the fact that $P_w x = 0$, together with (a).

(c) For any bilinear form, $A \in \mathbb{S} \Longrightarrow A = \pm A^\star = \pm M^{-1}A^TM \Longrightarrow MA = \pm A^TM \Longrightarrow x^TMA = \pm x^TA^TM = \pm(Ax)^TM$. But $Ax = 0$. Hence $x^TMA = 0$. From (2.2), we have $P_w = I - xw^TM$. Hence $AP_w = A - Axw^TM = A$, since $Ax = 0$. Using (2.1) and $x^TMA = 0$ we now obtain

$$P_w^\star A\, P_w = P_w^\star A = (I - wx^TM^T)A = A,$$

since for orthosymmetric bilinear forms $M^T = \pm M$. The proof for sesquilinear forms follows along the same lines. ◻

The complete solution to the *homogeneous mapping problem* can now be described.

THEOREM 4.2 (characterization for $\mathbb{J}$ and $\mathbb{L}$: homogeneous case). *Let $\mathbb{S}$ denote the Lie or Jordan algebra of any orthosymmetric scalar product space. Given $x \in \mathbb{K}^n$ with $x \neq 0$, and $w \in \mathbb{K}^n$ such that $\langle w, x\rangle_{\mathrm{M}} = 1$,*

$$\{A \in \mathbb{S} : Ax = 0\} = \{P_w^\star S P_w : S \in \mathbb{S}\}$$

*where $P_w$ is defined in* (2.2)*.*

*Proof.* The proof follows immediately by combining (b) and (c) of Lemma 4.1. ◻

COROLLARY 4.3. *If $v$, $w \in \mathbb{K}^n$, with $\langle v, x\rangle_{\mathrm{M}} = \langle w, x\rangle_{\mathrm{M}} = 1$, then*

$$\{P_v^\star S P_v : S \in \mathbb{S}\} = \{P_w^\star S P_w : S \in \mathbb{S}\}.$$

Thus we have several representations of the set of solutions to the homogeneous mapping problem. Now if $A$, $B \in \mathbb{S}$ are such that $Ax = Bx = b$, then $(A - B)x = 0$. By Theorem 4.2, $A - B = P_w^\star S P_w$, or equivalently, $A = B + P_w^\star S P_w$ for some $S \in \mathbb{S}$. Hence,

$$\{A \in \mathbb{S} : Ax = b\} = B + \{A \in \mathbb{S} : Ax = 0\}, \tag{4.1}$$

where $B$ is any particular solution of the nonhomogeneous mapping problem. By combining Theorems 3.2 and 4.2 together with (4.1) we now have the complete solution of the characterization part of the mapping problem for $\mathbb{J}$ and $\mathbb{L}$.

THEOREM 4.4 (characterization for $\mathbb{J}$ and $\mathbb{L}$: nonhomogeneous case). *Let $\mathbb{J}$ and $\mathbb{L}$ be the Jordan and Lie algebras of any orthosymmetric scalar product on $\mathbb{K}^n$. Let $x, b \in \mathbb{K}^n$ with $x \neq 0$ and let $\mathcal{J}$ and $\mathcal{L}$ be the structured mapping sets as in* (3.1)*. Choose any $v, w \in \mathbb{K}^n$ such that $\langle v, x\rangle_{\mathrm{M}} = \langle w, x\rangle_{\mathrm{M}} = 1$, and use $v$ and $w$ to define $P_v$, ${}^{\pm}B_w$ as in* (2.2) *and* (3.2)*, respectively. Consider the following sets:*

$$\mathcal{J}_+ = \{{}^{+}B_w + P_v^\star S P_v : S \in \mathbb{J}\}, \qquad \mathcal{L}_- = \{{}^{-}B_w + P_v^\star L P_v : L \in \mathbb{L}\}.$$

*Then*

$$\mathcal{J} = \begin{cases} \mathcal{J}_+ & \text{if } \langle x, b\rangle_{\mathrm{M}} = \langle b, x\rangle_{\mathrm{M}}, \\ \emptyset & \text{otherwise,} \end{cases} \qquad \mathcal{L} = \begin{cases} \mathcal{L}_- & \text{if } \langle x, b\rangle_{\mathrm{M}} = -\langle b, x\rangle_{\mathrm{M}}, \\ \emptyset & \text{otherwise.} \end{cases}$$

A more general problem for Hermitian, and real symmetric matrices in particular, was considered by Sun [17, Lem. 1.4]. For given matrices $X, B \in \mathbb{K}^{n\times \ell}$, Sun gave a characterization of the set

$$\mathcal{H} = \{A \in \mathbb{K}^{n\times n} : A^* = A \text{ and } AX = B\}$$

in terms of the pseudoinverse $X^+$ of $X$, and the complementary orthogonal projections $\Pi_X = XX^+$ and $\Pi_{X^\perp} = I - \Pi_X$. He proved that $\mathcal{H} \neq \emptyset$ if and only if two conditions are satisfied: $B\Pi_{X^*} = B$ and $\Pi_X B X^+$ is Hermitian. In this case $\mathcal{H}$ can be expressed as

$$\mathcal{H} = \{BX^+ + (BX^+)^*\Pi_{X^\perp} + \Pi_{X^\perp} S \Pi_{X^\perp} : \quad S^* = S,\ S \in \mathbb{K}^{n\times n}\}. \tag{4.2}$$

When $\ell = 1$, writing $X$, $B$ as $x$, $b$, respectively, we get $\Pi_x = xx^*/(x^*x)$, and $x^+ = x^*/(x^*x)$. Since $\Pi_{x^*} = 1$, the conditions for $\mathcal{H}$ to be nonempty reduce to requiring that

$\Pi_x b x^+$ be Hermitian. A simple calculation shows that this happens if and only if $x^*b$ is real, which is in agreement with the condition in Corollary 3.3. Sun's characterization of $\mathcal{H}$ becomes

$$\mathcal{H} = \left\{ \frac{bx^*}{x^*x} + \frac{x^*b}{x^*x}\Pi_x + \Pi_x S \Pi_x,\ S^* = S \right\},$$

which corresponds to $\mathcal{J}_+$ in Theorem 4.4 with $M = I$ and the special choice $v = w = x/(x^*x)$. This choice of $w$ corresponds to using an orthogonal projection in the representation for $\mathcal{J}_+$, since $P_v$ is now $I - xx^*/(x^*x)$. Thus Sun's characterization is one among many given by Theorem 4.4.

A similar analysis of the real symmetric case shows that the results of Corollary 3.3 and Theorem 4.4 are compatible with Sun's solution for the case $\ell = 1$, and due to the freedom in the choice of $v$ and $w$, give a more flexible description of the set of real symmetric matrices mapping $x$ to $b$.

## 5. Structured mappings with extremal properties.

Let $\mathcal{J}$ and $\mathcal{L}$ be the sets of all structured solutions to the mapping problem as in (3.1). We now show how to find matrices in $\mathcal{J}$ or $\mathcal{L}$ with the extremal properties of minimal rank, minimal Frobenius norm, or minimal 2-norm, and we investigate their uniqueness.

### 5.1. Structured mappings of minimal rank.

In what follows, we assume $b \neq 0$.

THEOREM 5.1 (rank-one structured mappings). *Let* $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ *be an orthosymmetric scalar product, and let* $\mathcal{S}$ *denote either* $\mathcal{J}$ *or* $\mathcal{L}$ *as in* (3.1). *Assume* $b \neq 0$. *A necessary condition for the existence of a rank-one matrix in* $\mathcal{S}$ *is* $\langle b, x\rangle_{\mathrm{M}} \neq 0$. *Whenever this rank-one matrix exists, it is unique and given by*

$$A = \begin{cases} bb^TM/\langle b,x\rangle_{\mathrm{M}} & \textit{for bilinear forms,} \\ bb^*M/\langle b,x\rangle_{\mathrm{M}} & \textit{for sesquilinear forms.} \end{cases}$$

*Proof.* Consider any rank-one matrix $A = uv^T$ such that $Ax = b$ with $b \neq 0$. Since $b \in \operatorname{range}(A)$, $u$ is a multiple of $b$, so without loss of generality we can take $u = b$.

Now suppose the orthosymmetric scalar product is bilinear, so $M = \pm M^T$. Since $M$ is nonsingular, there exists $z \in \mathbb{K}^n$ such that $v^T = z^TM$, and so $A = uv^T = bz^TM$. For $A \in \mathbb{S}$ we have $A^\star = \epsilon A$ with $\epsilon = \pm 1$. Hence by (2.1) we have $\pm zb^TM = \epsilon bz^TM$ and so $zb^T = \pm\epsilon bz^T$. Thus $z = \mu b$ and $A = \mu bb^TM$ with $\mu$ a scalar. But $Ax = b \Rightarrow \mu b(b^TMx) = b \Rightarrow \mu\langle b,x\rangle_{\mathrm{M}} = 1$, thus forcing $\langle b,x\rangle_{\mathrm{M}}$ to be nonzero, and uniquely determining $A$ by $A = bb^TM/\langle b,x\rangle_{\mathrm{M}}$. Similar reasoning applies for the sesquilinear case, leading to the formula $A = bb^*M/\langle b,x\rangle_{\mathrm{M}}$. ☐

COROLLARY 5.2. *Let* $b \neq 0$. *If* $\langle b,x\rangle_{\mathrm{M}} \neq 0$, *then either* $\mathcal{S}$ *is empty or there is a unique* $A \in \mathcal{S}$ *with* $\operatorname{rank}(A) = 1$.

*Proof.* When $\mathcal{S} \neq \emptyset$, we know from Theorem 3.2 that ${}^+B_w \in \mathcal{J}$ and ${}^-B_w \in \mathcal{L}$ for any $w$ such that $\langle w,x\rangle_{\mathrm{M}} = 1$. Since $\langle b,x\rangle_{\mathrm{M}} \neq 0$, choose $w = w_*$, where

$$w_* := \begin{cases} b/\langle b,x\rangle_{\mathrm{M}} & \text{for bilinear forms,} \\ b/\overline{\langle b,x\rangle}_{\mathrm{M}} & \text{for sesquilinear forms} \end{cases} \tag{5.1}$$

so that $\langle w_*,x\rangle_{\mathrm{M}} = 1$. Substituting this choice of $w$ into the formulas for ${}^+B_w$, ${}^-B_w$ given in (3.2) yields the unique rank-one mapping specified in Theorem 5.1. ☐

TABLE 5.1

*Rank-one structured mappings when* $\langle b, x\rangle_{\mathrm{M}} \neq 0$ *with* $w_*$ *as in* (5.1).

| Scalar product | $\mathcal{J}$ | $\mathcal{L}$ |
|---|---|---|
| Symmetric bilinear | ${}^{+}B_{w_*}$ | $\mathcal{L}$ is empty |
| Skew-symmetric bilinear | $\mathcal{J}$ is empty | ${}^{-}B_{w_*}$ |
| Hermitian sesquilinear | ${}^{+}B_{w_*}$ if $0 \neq \langle b, x\rangle_{\mathrm{M}} \in \mathbb{R}$.<br>Otherwise $\mathcal{J}$ is empty. | ${}^{-}B_{w_*}$ if $0 \neq \langle b, x\rangle_{\mathrm{M}} \in i\mathbb{R}$.<br>Otherwise $\mathcal{L}$ is empty. |
| Skew-Hermitian sesquilinear | ${}^{+}B_{w_*}$ if $0 \neq \langle b, x\rangle_{\mathrm{M}} \in i\mathbb{R}$.<br>Otherwise $\mathcal{J}$ is empty. | ${}^{-}B_{w_*}$ if $0 \neq \langle b, x\rangle_{\mathrm{M}} \in \mathbb{R}$.<br>Otherwise $\mathcal{L}$ is empty. |

Particular cases of Corollary 5.2 are summarized in Table 5.1. The extra conditions in the sesquilinear cases come from the results in Corollary 3.3.

For nonzero $b$ we have seen that the condition $\langle b, x\rangle_{\mathrm{M}} \neq 0$, while necessary for the existence of structured rank-one mappings, is precisely the condition that precludes the existence of *any* structured mappings in certain cases (see Theorem 3.2). On the other hand, Theorem 3.2 also shows that structured mapping sets $\mathcal{S}$ are never empty when the condition $\langle b, x\rangle_{\mathrm{M}} = 0$ is met. We turn to the question of determining what the minimal achievable rank is in this case.

THEOREM 5.3 (rank-two structured mappings). *Let* $\langle \cdot, \cdot\rangle_{\mathrm{M}}$ *be an orthosymmetric scalar product, and let* $\mathcal{S}$ *denote either* $\mathcal{J}$ *or* $\mathcal{L}$ *as in* (3.1)*. Consider any nonzero* $x$*,* $b \in \mathbb{K}^n$*. If* $\langle b, x\rangle_{\mathrm{M}} = 0$*, then*

$$\min_{A \in \mathcal{S}} \operatorname{rank}(A) = 2.$$

*There are always infinitely many matrices in* $\mathcal{S}$ *attaining this minimal rank. Among these are* ${}^{-}B_w \in \mathcal{L}$ *and* ${}^{+}B_w \in \mathcal{J}$*, where* ${}^{-}B_w$*,* ${}^{+}B_w$ *are given by* (3.2)*, with any choice of* $w \in \mathbb{K}^n$ *such that* $\langle w, x\rangle_{\mathrm{M}} = 1$*.*

*Proof.* If $\langle b, x\rangle_{\mathrm{M}} = 0$, then by Theorem 5.1, the minimum possible rank for matrices in $\mathcal{S}$ is 2. We know ${}^{+}B_w$, ${}^{-}B_w$ map $x$ to $b$ for all $w \in \mathbb{K}^n$ such that $\langle w, x\rangle_{\mathrm{M}} = 1$, and from Theorem 3.2 it follows that ${}^{+}B_w \in \mathbb{J}$ and ${}^{-}B_w \in \mathbb{L}$ for all such $w$. Since ${}^{+}B_w$, ${}^{-}B_w$ are at most rank two, and since they cannot be rank one, they are structured mappings of rank two. ◻

**5.2. Structured mappings of minimal Frobenius norm.** Another important special property is minimal norm, since this is directly related to structured backward errors for linear systems and eigenvalue problems [19], [20] as well as to the derivation of quasi-Newton methods [3]. We first consider minimal Frobenius norm; the minimal 2-norm case will be treated in the next section. For real symmetric or Hermitian matrices, it is well known [2], [3] that minimal Frobenius norm is achieved by

$$A_{\mathrm{opt}} = \frac{bx^* + xb^*}{x^*x} - \frac{(b^*x)}{(x^*x)^2}xx^*.$$

We show how to generalize this result to all Lie and Jordan algebras associated with scalar products that are both orthosymmetric and unitary. To prove the uniqueness of the structured mapping of minimal Frobenius norm, we will need the next two lemmas.

LEMMA 5.4. *In any real or complex inner product space, the associated norm* $\|\cdot\|$ *is strictly convex on independent vectors, that is,*

$$\|tu + (1-t)v\| \;<\; t\|u\| + (1-t)\|v\|\,, \quad 0 < t < 1\,,$$

*for any linearly independent* $u$ *and* $v$.

*Proof.* The Cauchy–Schwarz inequality implies that $\langle u, v\rangle + \langle v, u\rangle < 2\|u\|\|v\|$ for linearly independent $u, v$. A straightforward calculation then establishes the result. □

LEMMA 5.5. *For* $b \neq 0$, *the Frobenius norm is strictly convex on* $\mathcal{S}$ ($\mathcal{S} = \mathcal{J}, \mathcal{L}$).

*Proof.* Assuming $b \neq 0$, distinct $A, B \in \mathcal{S}$ are linearly independent. Since the Frobenius norm arises from the inner product $\langle A, B\rangle = \mathrm{tr}(A^*B)$, the result is immediate from Lemma 5.4. □

As Lemma 3.1 and Theorem 3.2 show, whenever the set of structured mappings $\mathcal{S}$ is nonempty, we can construct a parametrized set of structured maps ${}^{+}B_w$ or ${}^{-}B_w$ that take $x$ to $b$. The next theorem shows how this latitude in the choice of the parameter $w \in \mathbb{K}^n$, $\langle w, x\rangle_{\mathrm{M}} = 1$, can be exploited to identify the unique map of minimal Frobenius norm.

THEOREM 5.6 (minimal Frobenius norm structured mapping). *Let* $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ *be a scalar product that is both orthosymmetric and unitary. Let* $\mathcal{S}$ *denote either* $\mathcal{J}$ *or* $\mathcal{L}$ *as in* (3.1). *If* $\mathcal{S} \neq \emptyset$, *the problem*

$$\min_{A\in\mathcal{S}} \|A\|_F$$

*has a unique solution given by*

$$A_{\mathrm{opt}} = \frac{bx^*}{x^*x} + \epsilon\left(\frac{bx^*}{x^*x}\right)^{\star}\left(I - \frac{xx^*}{x^*x}\right), \qquad \epsilon = \begin{cases} 1 & \text{if } \mathcal{S} = \mathcal{J}, \\ -1 & \text{if } \mathcal{S} = \mathcal{L}. \end{cases} \tag{5.2}$$

*Moreover,*

$$\|A_{\mathrm{opt}}\|_F^2 = 2\frac{\|b\|_2^2}{\|x\|_2^2} - \beta^2\frac{|\langle b, x\rangle_{\mathrm{M}}|^2}{\|x\|_2^4}, \tag{5.3}$$

*where* $\beta > 0$ *is such that* $\beta M$ *is unitary.*

*Proof.* Since $\mathcal{S} \neq \emptyset$, we know from Theorem 3.2 that ${}^{+}B_w \in \mathcal{J}$ and ${}^{-}B_w \in \mathcal{L}$ for any $w \in \mathbb{K}^n$ such that $\langle w, x\rangle_{\mathrm{M}} = 1$. Choose $w = w_0$, where

$$w_0 = \begin{cases} M^{-T}\overline{x}/(x^*x) & \text{for bilinear forms,} \\ M^{-*}x/(x^*x) & \text{for sesquilinear forms.} \end{cases} \tag{5.4}$$

Then $\langle w_0, x\rangle_{\mathrm{M}} = 1$, and the expressions for the structured maps ${}^{\pm}B_{w_0}$ in (3.2) and the projection $P_{w_0}$ in (2.2) become

$${}^{\pm}B_{w_0} = \frac{bx^*}{x^*x} \pm \left(\frac{bx^*}{x^*x}\right)^{\star} P_{w_0}, \qquad P_{w_0} = I - \frac{xx^*}{x^*x}. \tag{5.5}$$

For brevity, let $A_0$ denote ${}^{\pm}B_{w_0}$, and let $P_0$ denote the orthogonal projection $P_{w_0}$. We now show that $A_0$ is the unique map of minimal Frobenius norm in $\mathcal{S}$.

Complete $\{\,x/\|x\|_2\}$ to an orthonormal basis $\{\,x/\|x\|_2, u_2, \ldots, u_n\}$ with respect to the standard inner product on $\mathbb{K}^n$. We first observe that for all $A \in \mathcal{S}$,

$$\|Ax\|_F = \|b\|_F = \|A_0x\|_F.$$

The characterization theorem, Theorem 4.4, with $v = w = w_0$ tells us that any $A \in \mathcal{S}$ can be written as $A = A_0 + P_0^\star S P_0$ for some $S \in \mathbb{S}$. Premultiplying $u_i$ by $A$ and taking the norm yields

$$\|Au_i\|_2^2 = \|A_0u_i\|_2^2 + \|P_0^\star SP_0u_i\|_2^2 + 2\,\mathrm{Re}\Big((A_0u_i)^*P_0^\star SP_0u_i\Big), \quad 2 \le i \le n. \tag{5.6}$$

When $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is unitary, the last term on the right-hand side of (5.6) always vanishes. To see this, first consider the case when the form is bilinear. Since the stars commute in a unitary scalar product and $x^*u_i = 0$, $i = 2\colon n$, we have

$$(A_0u_i)^* = \pm u_i^*\left(\left(\frac{bx^*}{x^*x}\right)^\star\right)^* = \pm u_i^*\left(\frac{xb^*}{x^*x}\right)^\star = \pm\left(\frac{u_i^*M^{-1}\bar{b}}{x^*x}\right)x^TM =: \alpha_i x^TM$$

and

$$(A_0u_i)^*P_0^\star SP_0u_i = \alpha_i x^TM\Big(M^{-1}\Big(I - \frac{\bar{x}x^T}{x^*x}\Big)M\Big)Su_i = \alpha_i(x^T - x^T)MSu_i = 0.$$

Similarly, for sesquilinear forms, $(A_0u_i)^* = \alpha_i x^*M$ with $\alpha_i = \pm(u_i^*M^{-1}b)/(x^*x)$ and

$$(A_0u_i)^*P_0^\star SP_0u_i = \alpha_i x^*M\Big(M^{-1}\Big(I - \frac{xx^*}{x^*x}\Big)M\Big)Su_i = \alpha_i(x^* - x^*)MSu_i = 0.$$

Therefore from (5.6), $\|Au_i\|_2 \ge \|A_0u_i\|_2$, $2 \le i \le n$. Recall that the Frobenius norm is unitarily invariant; since $\{\,x/\|x\|_2, u_2, \dots, u_n\}$ forms an orthonormal basis for $\mathbb{K}^n$,

$$\|A\|_F^2 = \frac{\|Ax\|_2^2}{\|x\|_2^2} + \sum_{i=2}^n \|Au_i\|_2^2 \ge \frac{\|A_0x\|_2^2}{\|x\|_2^2} + \sum_{i=2}^n \|A_0u_i\|_2^2 = \|A_0\|_F^2 \quad \forall A \in \mathcal{S},$$

showing that $A_0$ has minimal Frobenius norm.

It is well known that strictly convex functions have at most one minimizer [1, p. 4]. Therefore Lemma 5.5 implies that $A_0$ is unique for $b \ne 0$. When $b = 0$, $A_0 \equiv 0$ is clearly unique. Thus $A_{\mathrm{opt}}$, the unique structured map of minimal Frobenius norm, is $A_0$, defined by (5.2).

Finally, for the Frobenius norm of $A_{\mathrm{opt}}$ we have

$$\|A_{\mathrm{opt}}\|_F^2 = \Big(\|bx^*\|_F^2 + \|(bx^*)^\star P_0\|_F^2 + 2\epsilon\,\mathrm{Re}\,\big(\mathrm{tr}[xb^*(bx^*)^\star P_0]\big)\Big)/\|x\|_2^4. \tag{5.7}$$

Now $P_0x = 0$ implies

$$\mathrm{tr}[xb^*(bx^*)^\star P_0] = \mathrm{tr}[P_0xb^*(bx^*)^\star] = \mathrm{tr}(0) = 0. \tag{5.8}$$

Since $P_0$ is an orthogonal projection, $P_0^2 = P_0 = P_0^*$. Hence

$$\begin{aligned}\|(bx^*)^\star P_0\|_F^2 = \mathrm{tr}\left[(bx^*)^\star P_0\big((bx^*)^\star\big)^*\right] &= \|(bx^*)^\star\|_F^2 - \|(bx^*)^\star x\|_2^2/(x^*x)\\ &= \|(bx^*)\|_F^2 - x^*(bb^*)^\star x.\end{aligned} \tag{5.9}$$

For the last equality we have used the fact that $\|X^\star\|_F = \|X\|_F$ for any unitary scalar product.[1] Recall that $\beta M$ is unitary for some $\beta > 0$. Thus $M^{-1} = \beta^2M^*$ and

$$x^*(bb^*)^\star x = \beta^2\left\{\begin{array}{ll} x^*M^*\bar{b}b^TMx & \text{(bilinear forms)}\\ x^*M^*bb^*Mx & \text{(sesquilinear forms)}\end{array}\right\} = \beta^2|\langle b,x\rangle_{\mathrm{M}}|^2. \tag{5.10}$$

Now combining (5.8)–(5.10) into (5.7) gives the desired formula for $\|A_{\mathrm{opt}}\|_F^2$. $\square$

---

[1] Surprisingly, this property characterizes unitary scalar products [13].

**5.3. Structured mappings of minimal 2-norm.** From $Ax = b$ it is clear that $\|b\|_2/\|x\|_2$ is always a lower bound for $\|A\|_2$. For a large class of scalar products Theorem 5.6 also yields an upper bound:

$$\frac{\|b\|_2}{\|x\|_2} \le \min_{A\in\mathcal{S}} \|A\|_2 \le \min_{A\in\mathcal{S}} \|A\|_F \le \sqrt{2}\,\frac{\|b\|_2}{\|x\|_2}\,, \tag{5.11}$$

where $\mathcal{S}$ denotes either $\mathcal{J}$ or $\mathcal{L}$ as in (3.1). In this section we show that the lower bound is actually attained in any Lie or Jordan algebra of a scalar product that is both orthosymmetric and unitary.

Unlike the structured mapping of minimal Frobenius norm, mappings of minimal 2-norm in $\mathcal{S}$ are almost never unique. For example, consider the Jordan algebra of $n \times n$ symmetric matrices with $n \ge 3$, and take $x = e_1$ and $b = e_2$ to be the first and second columns of the identity matrix, respectively. Then all matrices of the form $A = \operatorname{diag}\left(\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right], S\right)$ with $S$ symmetric and $\|S\|_2 \le 1$ satisfy $A^T = A$, $Ax = b$ and have $\|A\|_2 = \|b\|_2/\|x\|_2 = 1$. Indeed, this formula captures *all* symmetric matrices mapping $e_1$ to $e_2$ that have minimal 2-norm. We develop similar characterizations of the sets of structured mappings of minimal 2-norm for large classes of Lie and Jordan algebras by reducing to the characterization problem for the following special structures:

$$\begin{aligned} \operatorname{Sym}(n,\mathbb{K}) &= \{A \in \mathbb{K}^{n\times n} : A^T = A\},\\ \operatorname{Skew}(n,\mathbb{K}) &= \{A \in \mathbb{K}^{n\times n} : A^T = -A\},\\ \operatorname{Herm}(n,\mathbb{C}) &= \{A \in \mathbb{C}^{n\times n} : A^* = A\}. \end{aligned} \tag{5.12}$$

We will use the simplified notation $\operatorname{Sym}(\mathbb{K})$, etc., when the size of the matrices is clear from the context. The technical details for these special structures can be found in the appendix. Recall that for nonzero $\mu \in \mathbb{K}$,

$$\operatorname{sign}(\mu) := \overline{\mu}/|\mu|.$$

THEOREM 5.7 (minimal 2-norm structured mappings: general case). *Let $\mathbb{S}_n$ be the Lie algebra $\mathbb{L}$ or Jordan algebra $\mathbb{J}$ of a scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ on $\mathbb{K}^n$ that is both orthosymmetric and unitary, so that $M \cdot \mathbb{S}_n$ is either* $\operatorname{Sym}(n,\mathbb{K})$, $\operatorname{Skew}(n,\mathbb{K})$, *or* $\gamma\operatorname{Herm}(n,\mathbb{C})$ *for some $|\gamma| = 1$. Also let $x, b \in \mathbb{K}^n \setminus \{0\}$ be vectors such that $\mathcal{S} = \{A \in \mathbb{S}_n : Ax = b\}$ is nonempty. Then*

$$\min_{A\in\mathcal{S}} \|A\|_2 = \frac{\|b\|_2}{\|x\|_2}.$$

*Furthermore, with $\mathcal{M} := \big\{A \in \mathcal{S} : \|A\|_2 = \|b\|_2/\|x\|_2\big\}$, there exists a unitary matrix $U$ such that*

$$\mathcal{M} = \left\{ \frac{\|b\|_2}{\|x\|_2} U^{\star}(\beta M)^{-1} \begin{bmatrix} R & 0 \\ 0 & S \end{bmatrix} U : S \in \widetilde{\mathbb{S}}_{n-r}, \|S\|_2 \le 1 \right\}, \tag{5.13}$$

*where $\beta > 0$ is a real constant such that $\beta M$ is unitary and $\star$ denotes the adjoint of the scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$. The number $r$, the structured class $\widetilde{\mathbb{S}}_{n-r}$, and $R \in \mathbb{K}^{r\times r}$ are given in each case by the following:*

(i) $M\cdot\mathbb{S}_n = \mathrm{Sym}(n,\mathbb{K})$: *$r = 1$ ($r = 2$) if $x$ and $\overline{Mb}$ are linearly dependent (independent), $\widetilde{\mathbb{S}}_{n-r} = \mathrm{Sym}(n-r,\mathbb{K})$, and*

$$R = \begin{cases} \mathrm{sign}(\mu) & \textit{if } \beta\overline{Mb} = \mu x \textit{ for some } \mu \in \mathbb{K}, \\ \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] & \textit{otherwise.} \end{cases}$$

(ii) $M\cdot\mathbb{S}_n = \mathrm{Skew}(n,\mathbb{K})$: *$r = 2$, $\widetilde{\mathbb{S}}_{n-r} = \mathrm{Skew}(n-2,\mathbb{K})$, and $R = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$.*

(iii) $M\cdot\mathbb{S}_n = \gamma\mathrm{Herm}(n,\mathbb{C})$ *for some $|\gamma| = 1$: $r = 1$ ($r = 2$) if $x$ and $Mb$ are linearly dependent (independent), $\widetilde{\mathbb{S}}_{n-r} = \gamma\mathrm{Herm}(n-r,\mathbb{C})$, and*

$$R = \begin{cases} \gamma\,\mathrm{sign}(\mu) & \textit{if } \gamma^{-1}\beta Mb = \mu x \textit{ for some } \mu \in \mathbb{R}, \\ \gamma\left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] & \textit{otherwise.} \end{cases}$$

*The matrix $U$ can be taken as the product of at most two unitary Householder reflectors; when $\mathbb{K} = \mathbb{R}$, $U$ is real orthogonal.*

*Proof.* For orthosymmetric $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ it is shown in [12, Thm. 8.4] that left multiplication by the matrix $M$ defining the scalar product is a bijection from $\mathbb{K}^{n\times n}$ to $\mathbb{K}^{n\times n}$ that maps $\mathbb{L}$ and $\mathbb{J}$ to Skew($\mathbb{K}$) and Sym($\mathbb{K}$) for bilinear forms, and to unit scalar multiples of Herm($\mathbb{C}$) for sesquilinear forms. Furthermore $\beta M$ being unitary implies that the map $\mathbb{S} \longmapsto \beta M\cdot\mathbb{S}$ is a 2-norm-preserving bijection. For bilinear forms, the equivalence of the equations $Ax = b$ and $\widetilde{A}x := (\beta MA)x = (\beta Mb) =: \widetilde{b}$ thus reduces the structured mapping problem for $\mathcal{S} = \{A \in \mathbb{S}_n : Ax = b\}$ in a 2-norm-preserving way to the structured mapping problem for finding $\widetilde{A}$ in Skew$(n,\mathbb{K})$ or Sym$(n,\mathbb{K})$ such that $\widetilde{A}x = \widetilde{b}$. Similarly for sesquilinear forms, the equivalence of $Ax = b$ and $\widetilde{A}x := (\gamma^{-1}\beta MA)x = (\gamma^{-1}\beta Mb) =: \widetilde{b}$ gives a 2-norm-preserving reduction of the structured mapping problem for $\mathcal{S}$ to that of finding $\widetilde{A}$ in Herm$(n,\mathbb{C})$ such that $\widetilde{A}x = \widetilde{b}$.

The value of $\min_{A\in\mathcal{S}} \|A\|_2$ and the formula for $\mathcal{M}$ in (5.13) now follow by applying Theorem A.2 to the minimal 2-norm structured mapping problem for $\widetilde{A}x = \widetilde{b}$, and then using the correspondence between $\widetilde{A}$ and $A$. ☐

Note the structure of formula (5.13) and how it automatically produces matrices in $\mathbb{S}_n$. In all cases, $\frac{\|b\|_2}{\|x\|_2}(\beta M)^{-1}\,\mathrm{diag}(R,S)$ is in $\mathbb{S}_n$, since the scalar product is orthosymmetric and $\|b\|_2/\|x\|_2$ and $\beta$ are real. Lemma 4.1(a) shows that $\star$-congruence preserves $\mathbb{L}$ and $\mathbb{J}$ structure, so $U^\star\frac{\|b\|_2}{\|x\|_2}(\beta M)^{-1}\,\mathrm{diag}(R,S)U$ is again in $\mathbb{S}_n$.

### 5.4. Comparison of the various "minimal" structured mappings.

We conclude section 5 by exploring the relationships between the three types of extremal mappings—minimal rank, Frobenius norm, and 2-norm—under the assumption that the scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ is both unitary and orthosymmetric.

In general the minimal Frobenius norm solution $A_{\mathrm{opt}}$ differs from the minimal rank solution. The latter is usually rank one, whereas $A_{\mathrm{opt}}$ is generally rank two. From (5.2) we see that $A_{\mathrm{opt}}$ is rank one if and only if $M^{-1}\bar{x} \in \mathrm{span}\{b\}$ for bilinear forms or $M^{-1}x \in \mathrm{span}\{b\}$ for sesquilinear forms.

For structured mappings of the minimal 2-norm, the following corollary of Theorem 5.7 singles out the unique matrix of minimal rank as well as that of minimal Frobenius norm.

Corollary 5.8. *Under the hypotheses of Theorem 5.7, let $\mathcal{M}$ denote the set of all minimal 2-norm mappings in $\mathcal{S} = \{A \in \mathbb{S} : Ax = b\}$. Assume further that $x, b$ are* *vectors such that $\mathcal{S}$ is nonempty. Consider the particular mapping*

$$A_2 := \frac{\|b\|_2}{\|x\|_2} U^\star (\beta M)^{-1} \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} U \in \mathcal{M}\,, \tag{5.14}$$

*obtained by setting $S$ equal to* 0 *in* (5.13). *Then $A_2$ is the unique solution of both the minimal rank problem $\min_{A\in\mathcal{M}} \operatorname{rank}(A)$ and the minimal Frobenius norm problem $\min_{A\in\mathcal{M}} \|A\|_F$. Moreover, either*

(1) *$A_2$ has rank one and $\|A_2\|_F = \|b\|_2/\|x\|_2$, or*

(2) *$A_2$ has rank two and $\|A_2\|_F = \sqrt{2}\,\|b\|_2/\|x\|_2$.*

*Case* (1) *occurs when $x$ and $\overline{Mb}$ ($x$ and $Mb$) are linearly dependent and the scalar product is bilinear* (*sesquilinear*). (*Note that if $M\cdot\mathbb{S} = \operatorname{Skew}(n,\mathbb{K})$, then this linear dependence implies that $\mathcal{S}$ is empty.*) *Otherwise case* (2) *holds.*

Are there any conditions under which there is a structured mapping in $\mathcal{S}$ that simultaneously has all three extremal properties? The next result provides a complete answer to this question.

THEOREM 5.9. *Let $\mathbb{S}$ be the Lie or Jordan algebra of a scalar product $\langle\cdot,\cdot\rangle_{\mathrm{M}}$ that is both unitary and orthosymmetric. Assume that $x, b \in \mathbb{K}^n \setminus \{0\}$ are vectors such that $\mathcal{S} = \{A \in \mathbb{S} :\ Ax = b\}$ is nonempty and $A_2$ is the matrix defined* (5.14). *Then the unique minimal Frobenius norm mapping $A_{\mathrm{opt}} \in \mathcal{S}$ has both minimal* 2*-norm and minimal rank in $\mathcal{S}$ if and only if the pair of vectors $(x, b)$ satisfies either property* (a) *or property* (b) *below.*

(a) *$M^{-1}\bar{x} \in \operatorname{span}\{b\}$ for bilinear forms or $M^{-1}x \in \operatorname{span}\{b\}$ for sesquilinear forms. In this case*

$$A_{opt} = A_2 = \begin{cases} bb^T M/\langle b, x\rangle_{\mathrm{M}} & \textit{for bilinear forms,} \\ bb^* M/\langle b, x\rangle_{\mathrm{M}} & \textit{for sesquilinear forms} \end{cases}$$

*is the unique rank-one mapping in $\mathcal{S}$.*

(b) *$\langle b, x\rangle_{\mathrm{M}} = 0$. In this case*

$$A_{\mathrm{opt}} = A_2 = \frac{bx^*}{x^*x} + \epsilon \left(\frac{bx^*}{x^*x}\right)^\star, \quad \epsilon = \begin{cases} 1 & \textit{if } \mathbb{S} = \mathbb{J}, \\ -1 & \textit{if } \mathbb{S} = \mathbb{L}, \end{cases}$$

*is the unique rank-two mapping in $\mathcal{M} = \{A \in \mathcal{S} :\ \|A\|_2 = \min_{B\in\mathcal{S}} \|B\|_2\}$.*

*Proof.* ($\Rightarrow$) $A_{\mathrm{opt}}$ having minimal 2-norm in $\mathcal{S}$ means that $A_{\mathrm{opt}} \in \mathcal{M}$, with minimal Frobenius norm in $\mathcal{M}$; thus $A_{\mathrm{opt}} = A_2$ by Corollary 5.8. But $A_2$ is either rank one or rank two. $A_2$ with rank one means $A_{\mathrm{opt}}$ has rank one, and therefore property (a) holds by the remarks preceding the corollary. On the other hand $A_2$ with rank two implies $\|A_{\mathrm{opt}}\|_F = \|A_2\|_F = \sqrt{2}\,\|b\|_2/\|x\|_2$, which by (5.3) implies that property (b) holds.

($\Leftarrow$) Property (a) implies that $A_{\mathrm{opt}}$ is rank one by the remarks preceding the corollary. But property (a) is equivalent to the linear dependence of $x$ and $\overline{Mb}$ ($x$ and $Mb$) for bilinear (sesquilinear) forms, which are precisely the conditions in Corollary 5.8 which guarantee that $A_2$ is rank one. The uniqueness of rank-one mappings in $\mathcal{S}$ from Theorem 5.1 now implies that $A_{\mathrm{opt}} = A_2$ has all three minimality properties.

Property (b) implies that $\|A_{\mathrm{opt}}\|_F = \sqrt{2}\,\|b\|_2/\|x\|_2$ by (5.3), and that the minimal rank in $\mathcal{S}$ is two by Theorem 5.3. By Corollary 5.8 we know that $\|A_2\|_F \le \sqrt{2}\,\|b\|_2/\|x\|_2$, so the uniqueness of minimal Frobenius norm mappings implies that $A_{\mathrm{opt}} = A_2$. This map has minimal rank two by case (2) of Corollary 5.8. $\square$

## 6. Concluding remarks.

In this paper we have presented complete, unified, and explicit solutions of the existence and characterization problems for structured mappings coming from Lie and Jordan algebras associated with orthosymmetric scalar products. In addition, in the set $\{ A \in \mathbb{S} : \; Ax = b \}$ we have identified and characterized the structured mappings of minimal rank, minimal Frobenius norm, and minimal 2-norm. These results have already found application in the analysis of structured condition numbers and backward errors [7], [20], and constitute the first step towards characterizing the set $\{ A \in \mathbb{S} : \; AX = B \}$, where $X$ and $B$ are given matrices.

In part II of this paper [14] we consider the same structured mapping problems for a third class of structured matrices $\mathbb{S}$ associated with a scalar product: the automorphism group $\mathbb{G}$ defined by

$$\mathbb{G} = \left\{ A \in \mathbb{K}^{n\times n} : \langle Ax, Ay \rangle_{\mathrm{M}} = \langle x, y \rangle_{\mathrm{M}} \; \forall x, y \in \mathbb{K}^n \right\} = \left\{ A \in \mathbb{K}^{n\times n} : A^\star = A^{-1} \right\}.$$

Unlike the corresponding Lie and Jordan algebras, the group $\mathbb{G}$ is a *nonlinear* subset of $\mathbb{K}^{n\times n}$; hence different (and somewhat more elaborate) techniques are needed to solve the structured mapping problems for $\mathbb{G}$. There are, however, some ways in which the results for groups $\mathbb{G}$ are actually simpler than the ones developed in this paper for $\mathbb{L}$ and $\mathbb{J}$. Consider, for example, the solution of the existence problem given in [14]: for any orthosymmetric scalar product $\langle \cdot, \cdot \rangle_{\mathrm{M}}$, there exists $A \in \mathbb{G}$ such that $Ax = b$ if and only if $\langle x, x \rangle_{\mathrm{M}} = \langle b, b \rangle_{\mathrm{M}}$. A clean, unified, and simply stated result. Examples of groups $\mathbb{G}$ covered by this theorem include the orthogonal, symplectic, and pseudounitary groups. Two types of characterization of the set $\{A \in \mathbb{G} : \; Ax = b\}$ are also given in [14], both of which are expected to be useful in structured backward error investigations.

## Appendix. Structured mappings of minimal 2-norm for symmetric, skew-symmetric, and Hermitian structures.

Our goal in this appendix is to characterize the complete set of all minimal 2-norm mappings for each of the five key structures in (5.12). For example, for real symmetric matrices it is already well known that $A = (\|b\|_2/\|x\|_2)H$, where $H$ is a Householder reflector mapping $x/\|x\|_2$ to $b/\|b\|_2$, provides a minimal 2-norm solution. However, the set of *all* minimal 2-norm symmetric matrices taking $x$ to $b$ has not previously been explicitly described.

First we consider the $2 \times 2$ case for a special type of $(x, b)$ vector pair and for symmetric and Hermitian structures.

LEMMA A.1. *Let* $\mathbb{S}$ *be either* $\mathrm{Sym}(2, \mathbb{K})$ *or* $\mathrm{Herm}(2, \mathbb{C})$ *and let*

$$\mathcal{S} = \left\{ A \in \mathbb{S} : \; A \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix} \right\},$$

*where* $\alpha, \beta \in \mathbb{C}$ *with* $\mathrm{Re}(\alpha) \neq 0$ *and* $\beta \neq 0$ *when* $\mathbb{S} = \mathrm{Sym}(2, \mathbb{C})$*, and* $\alpha, \beta \in \mathbb{R} \setminus \{0\}$ *otherwise. Then*

$$\min_{A \in \mathcal{S}} \|A\|_2 = 1,$$

*with* $A = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$ *being the* unique *matrix in* $\mathcal{S}$ *of minimal* 2*-norm.*

*Proof.* Note that from (5.11) any $A \in \mathcal{S}$ satisfies $\|A\|_2 \geq 1$, and since $\left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] \in \mathcal{S}$ has unit 2-norm we have $\min_{A \in \mathcal{S}} \|A\|_2 = 1$. The rest of the proof consists of showing that $\left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$ is the unique minimizer of the 2-norm for $\mathcal{S}$.

We start by parameterizing $\mathcal{S}$ using (4.1):

$$\mathcal{S} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} + \left\{ A \in \mathbb{S} : \; A \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\},$$

where $\left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$ is a particular mapping in $\mathcal{S}$. Any $A \in \mathrm{Sym}(2,\mathbb{K})$ has the form $\left[\begin{smallmatrix}a & z\\ z & c\end{smallmatrix}\right]$ with $a, c, z \in \mathbb{K}$, so $A\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right] = \left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right]$ implies $\left[\begin{smallmatrix}a & z\\ z & c\end{smallmatrix}\right] = z\left[\begin{smallmatrix}-\beta/\alpha & 1\\ 1 & -\alpha/\beta\end{smallmatrix}\right]$. Similarly any $A \in \mathrm{Herm}(2,\mathbb{C})$ has the form $\left[\begin{smallmatrix}a & z\\ \bar{z} & c\end{smallmatrix}\right]$ with $a, c \in \mathbb{R}$; then $\alpha, \beta \in \mathbb{R}$ together with $A\left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right] = \left[\begin{smallmatrix}0\\ 0\end{smallmatrix}\right]$ implies that $z \in \mathbb{R}$, and so $A$ can once again be expressed in the form $z\left[\begin{smallmatrix}-\beta/\alpha & 1\\ 1 & -\alpha/\beta\end{smallmatrix}\right]$. Hence writing

$$P(z) = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} + z\begin{bmatrix}-\frac{\beta}{\alpha} & 1\\ 1 & -\frac{\alpha}{\beta}\end{bmatrix} = \begin{bmatrix}1 - \frac{\beta}{\alpha}z & z\\ z & -1 - \frac{\alpha}{\beta}z\end{bmatrix},$$

we have $\mathcal{S} = \{P(z) : \ z \in \mathbb{K}\}$ if $\mathbb{S} = \mathrm{Sym}(2,\mathbb{K})$, and $\mathcal{S} = \{P(z) : \ z \in \mathbb{R}\}$ when $\mathbb{S} = \mathrm{Herm}(2,\mathbb{C})$ and $\alpha, \beta \in \mathbb{R}$.

We can now calculate the 2-norm of $P(z)$ by computing the largest eigenvalue of

$$P^*P(z) = \begin{bmatrix}1 - \left(\frac{\beta}{\bar{\alpha}}\bar{z} + \frac{\beta}{\alpha}z\right) + \left(1 + \frac{\beta^2}{|\alpha|^2}\right)|z|^2 & (z - \bar{z}) - \gamma|z|^2\\ (\bar{z} - z) - \bar{\gamma}|z|^2 & 1 + \left(\frac{\bar{\alpha}}{\beta}\bar{z} + \frac{\alpha}{\beta}z\right) + \left(1 + \frac{|\alpha|^2}{\beta^2}\right)|z|^2\end{bmatrix},$$

where $\gamma := (\alpha/\beta) + (\beta/\overline{\alpha})$. Much calculation and simplification yields

$$\operatorname{tr} P^*P(z) = 2 + 2q(z), \qquad \det P^*P(z) = 1 + 2q(z) - 2\Big(1 + \operatorname{Re}\frac{\alpha^2}{|\alpha|^2}\Big)|z|^2,$$

where $q(z) := \operatorname{Re}\left[\left(\frac{\alpha}{\beta} - \frac{\beta}{\alpha}\right)z\right] + |\gamma|^2|z|^2/2 \in \mathbb{R}$. Since the characteristic polynomial of $P^*P(z)$ is $\lambda^2 - \operatorname{tr} P^*P(z)\lambda + \det P^*P(z)$, we get

$$\begin{aligned}\lambda_{\pm}(z) &= \frac{1}{2}\left(\operatorname{tr} P^*P(z) \pm \sqrt{\Big(\operatorname{tr} P^*P(z)\Big)^2 - 4\det P^*P(z)}\ \right)\\ &= 1 + q(z) \pm \sqrt{q(z)^2 + 2\Big(1 + \operatorname{Re}\frac{\alpha^2}{|\alpha|^2}\Big)|z|^2}\ .\end{aligned}$$

Since $q(z) \in \mathbb{R}$, clearly the largest eigenvalue of $P^*P(z)$ is $\lambda_+(z)$. But the hypothesis $\operatorname{Re}(\alpha) \neq 0$ means that $\operatorname{Re}\frac{\alpha^2}{|\alpha|^2} > -1$, so the second term under the square root is strictly bigger than 0 for all nonzero $z$. Hence $\lambda_+(z)$ satisfies $\lambda_+(0) = 1$ and $\lambda_+(z) > 1$ for all nonzero $z$. Thus $z = 0$ is the unique minimizer of $\lambda_+(z)$, and hence $P(0) = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$ is the unique minimizer of the 2-norm for $\mathcal{S}$. $\square$

We are now in a position to give a complete description of the set of all minimal 2-norm structured mappings for symmetric, skew-symmetric, and Hermitian matrices.

THEOREM A.2 (minimal 2-norm structured mappings: special cases). *Let* $\mathbb{S}_n$ *be either* $\mathrm{Sym}(n,\mathbb{K})$, $\mathrm{Skew}(n,\mathbb{K})$, *or* $\mathrm{Herm}(n,\mathbb{C})$, *and let* $x, b \in \mathbb{K}^n \setminus \{0\}$ *be vectors such that* $\mathcal{S} = \{A \in \mathbb{S}_n : Ax = b\}$ *is nonempty. Then*

$$\min_{A \in \mathcal{S}} \|A\|_2 = \frac{\|b\|_2}{\|x\|_2}.$$

*Furthermore with* $\mathcal{M} := \big\{A \in \mathcal{S} : \|A\|_2 = \|b\|_2/\|x\|_2\big\}$, *there exists an* $n \times n$ *unitary matrix* $U$ *such that*

$$\mathcal{M} = \left\{\frac{\|b\|_2}{\|x\|_2}U^{\star}\operatorname{diag}(R, S)U : \ S \in \mathbb{S}_{n-r}, \ \|S\|_2 \le 1\right\},$$

*where the adjoint* $\star$, *the number* $r$, *and* $R \in \mathbb{S}_r$ *are given by the following:*

(i) $\mathbb{S}_n = \mathrm{Sym}(n, \mathbb{K})$: $\star = T$ *and* $r = 1$ $(r = 2)$ *if* $x$ *and* $\overline{b}$ *are linearly dependent* (*independent*)*, with*

$$R = \begin{cases} \mathrm{sign}(\mu) & \textit{if } \overline{b} = \mu x \textit{ for some } \mu \in \mathbb{K}, \\ \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] & \textit{otherwise.} \end{cases}$$

(ii) $\mathbb{S}_n = \mathrm{Skew}(n, \mathbb{K})$: $\star = T$ *and* $r = 2$*, with* $R = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right]$.

(iii) $\mathbb{S}_n = \mathrm{Herm}(n, \mathbb{C})$: $\star = *$ *and* $r = 1$ $(r = 2)$ *if* $x$ *and* $b$ *are linearly dependent* (*independent*)*, with*

$$R = \begin{cases} \mathrm{sign}(\mu) & \textit{if } b = \mu x \textit{ for some } \mu \in \mathbb{R}, \\ \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] & \textit{otherwise.} \end{cases}$$

*The matrix* $U$ *can be taken as the product of at most two unitary*[2] *Householder reflectors; when* $\mathbb{K} = \mathbb{R}$, $U$ *is real orthogonal.*

*Proof.* For any $A \in \mathbb{S}_n$ such that $Ax = b$, observe that the matrix $B = \frac{\|x\|_2}{\|b\|_2} A$ is also in $\mathbb{S}_n$ and maps $\frac{x}{\|x\|_2}$ to $\frac{b}{\|b\|_2}$; also note that $\overline{b} = \mu x$ (resp., $b = \mu x$) in the theorem is equivalent to $\frac{\overline{b}}{\|b\|_2} = \mathrm{sign}(\overline{\mu}) \frac{x}{\|x\|_2}$ $\big($resp., $\frac{b}{\|b\|_2} = \mathrm{sign}(\overline{\mu}) \frac{x}{\|x\|_2}\big)$. Thus it suffices to prove the theorem for $x, b \in \mathbb{K}^n$ with $\|x\|_2 = \|b\|_2 = 1$ and the condition $\overline{b} = \mu x$ (resp., $b = \mu x$) replaced by $\overline{b} = \mathrm{sign}(\overline{\mu}) x$ $\big($resp., $b = \mathrm{sign}(\overline{\mu}) x\big)$; we make these assumptions throughout the rest of the argument.

The proof proceeds by first developing a unitary $U$ and accompanying $R$ for each of the five structures in (5.12). Then these $U$ and $R$ matrices are used to build explicit families of matrices in the structured mapping set $\mathcal{S}$ that realize the lower bound in (5.11), and thus are of minimal 2-norm. Finally we show that for each structure these families account for all of $\mathcal{M}$.

We begin by constructing for each case of the theorem a unitary matrix $U$ such that

$$Ux = \begin{bmatrix} y \\ 0 \end{bmatrix}, \quad (U^\star)^{-1} b = \begin{bmatrix} c \\ 0 \end{bmatrix}, \tag{A.1}$$

with $y, c \in \mathbb{K}^r$ satisfying $Ry = c$, where $R \in \mathbb{S}_r$ is as defined in the theorem.

(i) First, suppose that $\mathbb{S}_n = \mathrm{Sym}(n, \mathbb{K})$. If $\overline{b} = \mathrm{sign}(\overline{\mu}) x$ for some $\mu \in \mathbb{K}$, then let $U$ be the unitary Householder reflector mapping $x$ to $e_1$, so that $y = 1$. Then $(U^\star)^{-1} b = \overline{U} b = \mathrm{sign}(\mu) e_1$, so $c = \mathrm{sign}(\mu)$. Clearly with $R := \mathrm{sign}(\mu) \in \mathbb{S}_1$ we have $Ry = c$.

When $x$ and $\overline{b}$ are linearly independent then $U$ can be taken as the product of two unitary Householder reflectors, $U = H_2 H_1$. The first reflector $H_1$ takes $x + \overline{b}$ to $\pm\| x + \overline{b} \|_2 e_1$; with $H_1 x = \left[\begin{smallmatrix} \alpha \\ v \end{smallmatrix}\right]$ and $H_1 \overline{b} = \left[\begin{smallmatrix} \gamma \\ w \end{smallmatrix}\right]$ we see that $w = -v$ with $v \neq 0$ because of the linear independence of $x$ and $\overline{b}$, and $\alpha + \gamma = \pm\| x + \overline{b} \|_2 \in \mathbb{R} \setminus \{0\}$. Then $\|x\|_2 = \| \overline{b} \|_2 \Rightarrow \|H_1 x\|_2 = \|H_1 \overline{b} \|_2 \Rightarrow |\alpha| = |\gamma|$, which together with $\alpha + \gamma \in \mathbb{R}$ implies that $\gamma = \overline{\alpha}$, and hence $H_1 \overline{b} = \left[\begin{smallmatrix} \overline{\alpha} \\ -v \end{smallmatrix}\right]$. Note also that $2 \,\mathrm{Re}\, \alpha = \alpha + \overline{\alpha} = \pm\| x + \overline{b} \|_2 \neq 0$. For the second reflector pick $H_2 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & \widetilde{H}_2 \end{smallmatrix}\right]$ so that $\widetilde{H}_2 v = \beta e_1$ with $\beta = \pm\|v\|_2 \neq 0$.

---

[2] But not necessarily Hermitian; see [11].

Hence

$$U\,[\,x \quad \overline{b}\,] = \begin{bmatrix} \alpha & \overline{\alpha} \\ \beta & -\beta \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}, \quad \operatorname{Re}\alpha \neq 0,\ 0 \neq \beta \in \mathbb{R}, \tag{A.2}$$

and therefore $y = \left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$ and $c = \left[\begin{smallmatrix}\alpha\\ -\beta\end{smallmatrix}\right]$ satisfy $Ry = c$ with $R = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right] \in \mathbb{S}_2$. Note that $U$ can be taken to be real orthogonal when $\mathbb{K} = \mathbb{R}$.

(ii) For $\mathbb{S}_n = \operatorname{Skew}(n, \mathbb{K})$, Theorem 3.2 says that $\mathcal{S}$ is nonempty if and only if $b^T x = 0$. In this situation $U$ can be taken as the product of two unitary Householder reflectors, $U = H_2 H_1$. The first reflector $H_1$ is defined to take $x$ to $e_1$; then $H_1 \overline{b} = \left[\begin{smallmatrix}\alpha\\ v\end{smallmatrix}\right]$ for some $v \in \mathbb{K}^{n-1}$. The fact that $b^T x = 0$ implies $\alpha = 0$, since $b^T x = [\,\overline{\alpha} \quad v^*\,] H_1 H_1^* e_1 = \overline{\alpha}$. For the second reflector pick $H_2 = \left[\begin{smallmatrix}1 & 0\\ 0 & \widetilde{H}_2\end{smallmatrix}\right]$ so that $\widetilde{H}_2 v = e_1 \in \mathbb{K}^{n-1}$. Then $U\,[\,x \quad \overline{b}\,] = [\,e_1 \quad e_2\,] \in \mathbb{K}^{n\times 2}$, giving $y = \left[\begin{smallmatrix}1\\ 0\end{smallmatrix}\right]$ and $c = \left[\begin{smallmatrix}0\\ 1\end{smallmatrix}\right]$ in $\mathbb{K}^2$ satisfying $Ry = c$ for $R = \left[\begin{smallmatrix}0 & -1\\ 1 & 0\end{smallmatrix}\right] \in \mathbb{S}_2$. Note once again that $U$ can be taken to be real orthogonal when $\mathbb{K} = \mathbb{R}$.

(iii) Finally suppose that $\mathbb{S}_n = \operatorname{Herm}(n, \mathbb{C})$. Theorem 3.2 says that $\mathcal{S}$ is nonempty if and only if $b^* x \in \mathbb{R}$. If $x$ and $b$ are linearly dependent, then $b = \operatorname{sign}(\overline{\mu})x$ for some $\mu \in \mathbb{C}$, and $b^* x \in \mathbb{R}$ implies that $\mu \in \mathbb{R}$. In this case $U$ can be taken as the unitary Householder reflector mapping $x$ to $e_1$ so that $(U^\star)^{-1} b = Ub = \operatorname{sign}(\mu) e_1$ since $\mu$ is real. Hence $[\,y \ \ c\,] = [\,1 \ \ \operatorname{sign}(\mu)\,]$ and $Ry = c$ with $R = \operatorname{sign}(\mu) \in \mathbb{S}_1$.

On the other hand if $x$ and $b$ are linearly independent, then $U$ can be taken as the product of two unitary Householder reflectors $U = H_2 H_1$ in a manner analogous to that described above in (i) for $\operatorname{Sym}(n, \mathbb{K})$; the only difference is that $H_1$ now takes $x + b$ to $\pm\|x + b\|_2 e_1$. In this case (A.2) holds with $\overline{b}$ replaced by $b$. Also $b^* x = (H_1 b)^*(H_1 x) = \alpha^2 - v^* v \in \mathbb{R}$ so that $\alpha^2 \in \mathbb{R}$. This together with $\operatorname{Re}\alpha \neq 0$ implies that $\alpha \in \mathbb{R}$. Hence we have $y = \left[\begin{smallmatrix}\alpha\\ \beta\end{smallmatrix}\right]$ and $c = \left[\begin{smallmatrix}\alpha\\ -\beta\end{smallmatrix}\right]$ with $\alpha, \beta \in \mathbb{R} \setminus \{0\}$, satisfying $Ry = c$ with $R = \left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right] \in \mathbb{S}_2$.

Using the unitary $U$ and $R \in \mathbb{S}_r$ constructed above for each $\mathbb{S}_n$, we can now show that the lower bound $1 = \|b\|_2 / \|x\|_2 \leq \min_{A \in \mathcal{S}} \|A\|_2$ from (5.11) is actually attained by a whole family of $A \in \mathcal{S}$. For any $S \in \mathbb{S}_{n-r}$ with $\|S\|_2 \leq 1$, consider $A = U^\star \operatorname{diag}(R, S) U$. Then $A \in \mathbb{S}_n$, since $\mathbb{S}_n$ is preserved by any $\star$-congruence (see Lemma 4.1(a)) and $\operatorname{diag}(R, S) \in \mathbb{S}_n$. Also $Ax = b$ because of the properties of $U$ in (A.1), and $\|A\|_2 = \|\operatorname{diag}(R, S)\|_2 = \|R\|_2 = 1$. Thus

$$\Big\{ U^\star \operatorname{diag}(R, S) U : \ S \in \mathbb{S}_{n-r}\,, \ \|S\|_2 \leq 1 \Big\} \subseteq \mathcal{M}\,.$$

Finally, we complete the characterization of $\mathcal{M}$ by showing that this containment is actually an equality. Consider an arbitrary $A \in \mathcal{M}$. Then $Ax = b \ \Rightarrow\ ((U^\star)^{-1} A U^{-1})(Ux) = (U^\star)^{-1} b$, so the matrix $B := (U^\star)^{-1} A U^{-1} = (U^{-1})^\star A U^{-1}$ is in $\mathbb{S}_n$ and maps the vector $Ux = \left[\begin{smallmatrix}y\\ 0\end{smallmatrix}\right]$ to $(U^\star)^{-1} b = \left[\begin{smallmatrix}c\\ 0\end{smallmatrix}\right]$. Let $B_{11} \in \mathbb{S}_r$ be the leading principal $r \times r$ submatrix of $B$, so $\|B_{11}\|_2 \leq \|B\|_2 = \|A\|_2 = 1$. The form of the two vectors $\left[\begin{smallmatrix}y\\ 0\end{smallmatrix}\right]$ and $\left[\begin{smallmatrix}c\\ 0\end{smallmatrix}\right]$ implies that $B_{11}$ maps $y$ to $c$; since $\|y\|_2 = \|c\|_2 = 1$ we have $\|B_{11}\|_2 \geq 1$, and hence $\|B_{11}\|_2 = 1$. Using Lemma A.1 we can now show that $B_{11} = R$ in all cases.

(i) Suppose $\mathbb{S}_n = \operatorname{Sym}(n, \mathbb{K})$ and $B_{11} \in \mathbb{S}_r$. Then $\overline{b} = \operatorname{sign}(\overline{\mu})x$ for some $\mu \in \mathbb{K}$ implies $[\,y \ c\,] = [\,1 \ \operatorname{sign}(\mu)\,]$, so $B_{11} y = c$ implies $B_{11} = \operatorname{sign}(\mu) = R$. On the other hand if $\bar{b}$ and $x$ are linearly independent, then $[\,y\ c\,] = \left[\begin{smallmatrix} \alpha & \alpha \\ \beta & -\beta \end{smallmatrix}\right]$ with $\mathrm{Re}(\alpha) \neq 0$ and $0 \neq \beta \in \mathbb{R}$. Since $B_{11}y = c$ with $\|B_{11}\|_2 = 1$, Lemma A.1 implies that $B_{11} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] = R$.

(ii) $B_{11} \in \mathrm{Skew}(2, \mathbb{K})$ must have the form $\left[\begin{smallmatrix} 0 & -\sigma \\ \sigma & 0 \end{smallmatrix}\right]$ for some $\sigma \in \mathbb{K}$. So $B_{11}y = c$ with $y = e_1$ and $c = e_2$ implies $\sigma = 1$, and hence $B_{11} = \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right] = R$.

(iii) Finally consider $\mathbb{S}_n = \mathrm{Herm}(n, \mathbb{C})$ and $B_{11} \in \mathbb{S}_r$. If $b = \mathrm{sign}(\mu)x$ for some $\mu \in \mathbb{R}$, then $[\,y\ c\,] = [\,1\ \ \mathrm{sign}(\mu)\,]$, so $B_{11}y = c$ implies $B_{11} = \mathrm{sign}(\mu) = R$. If $b$ and $x$ are linearly independent, then $[\,y\ c\,] = \left[\begin{smallmatrix} \alpha & \alpha \\ \beta & -\beta \end{smallmatrix}\right]$ with $\alpha, \beta$ both real and nonzero. Since $B_{11}y = c$ with $\|B_{11}\|_2 = 1$, Lemma A.1 implies that $B_{11} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right] = R$.

The condition $\|B\|_2 = 1$ now forces the rest of the first $r$ columns of $B$ to be all zeros; since $B \in \mathbb{S}_n$, the rest of the first $r$ rows of $B$ must also be all zeros. Thus $B$ has the form $B = \mathrm{diag}(R, S)$, with $S \in \mathbb{S}_{n-r}$. Finally, $\|B\|_2 = 1$ and $\|R\|_2 = 1$ implies that $\|S\|_2 \leq 1$. Thus we have $B := (U^\star)^{-1}AU^{-1} = \mathrm{diag}(R, S)$, so $A = U^\star \,\mathrm{diag}(R, S)U$ and hence $\mathcal{M} \subseteq \big\{U^\star \,\mathrm{diag}(R, S)U : S \in \mathbb{S}_{n-r}\,,\ \|S\|_2 \leq 1\big\}$. $\square$

Note that when $\mathbb{S}_n$ is the class of real symmetric matrices and if $x, y \in \mathbb{R}^n$ are linearly independent, then choosing $S = I_n$ in Theorem A.2 yields the Householder reflector $I - 2U^T e_2 e_2^T U$ mapping $x$ to $b$.

## REFERENCES

[1] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization. Theory and Examples*, Springer-Verlag, New York, 2000.

[2] J. R. Bunch, J. W. Demmel, and C. F. Van Loan, *The strong stability of algorithms for solving symmetric linear systems*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 494–499.

[3] J. E. Dennis, Jr. and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.

[4] D. J. Higham and N. J. Higham, *Backward error and condition of structured linear systems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 162–175.

[5] R. A. Horn, V. V. Sergeichuk, and N. Shaked-Monderer, *Solution of linear matrix equations in a *congruence class*, Electron. J. Linear Algebra, 13 (2005), pp. 153–156.

[6] C. R. Johnson and R. L. Smith, *Linear interpolation problems for matrix classes and a transformational characterization of M-matrices*, Linear Algebra Appl., 330 (2001), pp. 43–48.

[7] M. Karow, D. Kressner, and F. Tisseur, *Structured eigenvalue condition numbers*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1052–1068.

[8] C. G. Khatri and S. K. Mitra, *Hermitian and nonnegative definite solutions of linear matrix equations*, SIAM J. Appl. Math., 31 (1976), pp. 579–585.

[9] R.-W. Liu and R. J. Leake, *Exhaustive equivalence classes of optimal systems with separable controls*, SIAM J. Control, 4 (1966), pp. 678–685.

[10] M. Machover, *Matrices which take a given vector into a given vector*, Amer. Math. Monthly, 74 (1967), pp. 851–852.

[11] D. S. Mackey, N. Mackey, and F. Tisseur, *$\mathbb{G}$-reflectors: Analogues of Householder transformations in scalar product spaces*, Linear Algebra Appl., 385 (2004), pp. 187–213.

[12] D. S. Mackey, N. Mackey, and F. Tisseur, *Structured factorizations in scalar product spaces*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 821–850.

[13] D. S. Mackey, N. Mackey, and F. Tisseur, *On the Definition of Two Natural Classes of Scalar Product*, MIMS EPrint 2007.64, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2007.

[14] D. S. Mackey, N. Mackey, and F. Tisseur, *Structured Mapping Problems for Matrices Associated with Scalar Products, Part* II: *Automorphism Groups*, MIMS EPrint, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, in preparation.

[15] A. Pinkus, *Interpolation by matrices*, Electron. J. Linear Algebra, 11 (2004), pp. 281–291.

[16] S. M. Rump, *Structured perturbations part* I: *Normwise distances*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 1–30.

[17] J. Sun, *Backward perturbation analysis of certain characteristic subspaces*, Numer. Math., 65 (1993), pp. 357–382.

[18] J. Sun, *Backward Errors for the Unitary Eigenproblem*, Tech. report UMINF-97.25, Department of Computing Science, University of Umeå, Umeå, Sweden, 1997.

[19] F. Tisseur, *A chart of backward errors for singly and doubly structured eigenvalue problems*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 877–897.

[20] F. Tisseur and S. Graillat, *Structured condition numbers and backward errors in scalar product spaces*, Electron. J. Linear Algebra, 15 (2006), pp. 159–177.

[21] G. Trenkler, *Matrices which take a given vector into a given vector—revisited*, Amer. Math. Monthly, 111 (2004), pp. 50–52.

[22] Z.-Z. Zhang, X.-Y. Hu, and L. Zhang, *On the Hermitian-generalized Hamiltonian solutions of linear matrix equations*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 294–303.